Advanced Java Lab

Subject Code: MCAL12

A Practical Journal Submitted in Fulfillmentof the Degree of

MASTER IN COMPUTER APPLICATION

Year 2022-2023
By

(**Ravishankar Jaiswal**

(172047)

Semester- 1 Under the Guidance of

MS. Richa Ma'am

Institute of Distance and Open Learning

Vidya Nagari, Kalina, Santacruz East – 400098.

University of Mumbai

PCP Center

[Satish Pradhan Dyanasadhana College, Thane]

Institute of Distance and Open Learning,

Vidyanagari, Kalina, Santacruz (E) -400098

CERTIFICATE

This to certify that, (**Ravishankar Jaiswal)** appearing **Master in Computer Application (Semester I) (172047):** has satisfactory completed the prescribed practical of **MCAL12- Advanced JAVA Lab** as laid down by the University of Mumbai for the academic year 2022-23

Teacher in charge	Examiners	Coordinator IDOL, MCA University of Mumbai

Date: -

Place: -

Practical No: 1 Practical No: 1

Assignments on Java Generics

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

Code:

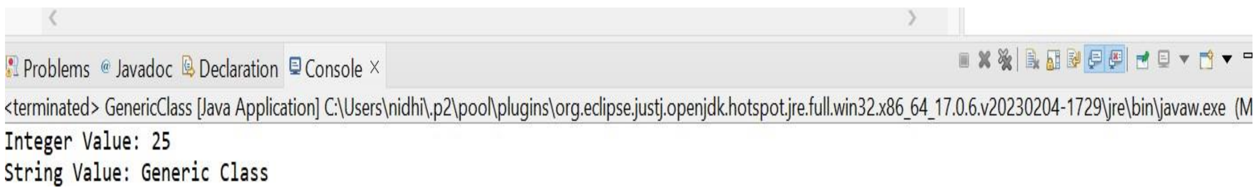
```
package Mappack;

public class GenericClass <T>{
private T t;
public void add(T t) {
this.t=t;
}
public T get() {
return t;
}
public static void main(String args[]) {
GenericClass<Integer> intObj =new GenericClass<Integer>();
GenericClass<String> strObj =new GenericClass<String>();

intObj.add(new ~~Integer~~(25));
strObj.add(new String("Generic Class"));
System.out.println("Integer Value: "+intObj.get());
System.out.println("String Value: "+strObj.get());
}

}
```

Output:



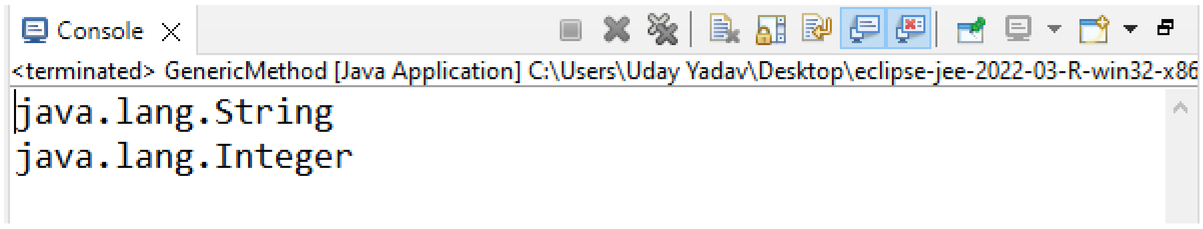
```
<terminated> GenericClass [Java Application] C:\Users\nidhi\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.6.v20230204-1729\jre\bin\javaw.exe  (M
Integer Value: 25
String Value: Generic Class
```

2. Write a Java Program to demonstrate Generic Methods.

Code:

```
package Mappack;
public class GenericMethod {
// TODO Auto-generated method stub
public static<T> void print(T t) {
System.out.println(t.getClass().getName());
}
public static void main(String args[]) {
GenericMethod.print("Hello World");
GenericMethod.print(100);
}
}
```

Output:



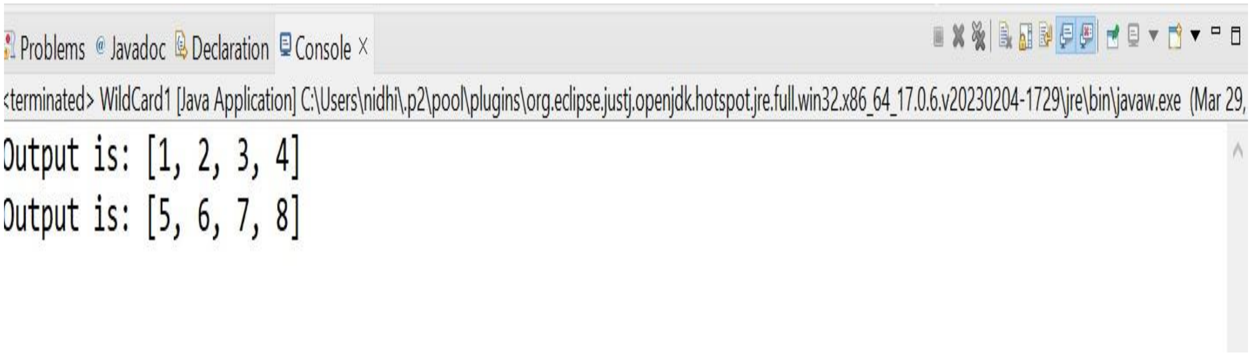
```
<terminated> GenericMethod [Java Application] C:\Users\Uday Yadav\Desktop\eclipse-jee-2022-03-R-win32-x86
java.lang.String
java.lang.Integer
```

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

```
package Mappack;
import java.util.Arrays;
import java.util.List;
public class WildCard1 {
public static void main(String args[]) {
//Lower bounded Integer List
List<Integer> list1 =Arrays.asList(1,2,3,4);
//Integer List object is being passed
print(list1);
//Lower bounded Number List
List<Number> list2= Arrays.asList(5,6,7,8);
print(list2);
}
public static void print(List<? super Integer> list) {
System.out.println("Output is: "+list);
}

}
```

Output:



```
<terminated> WildCard1 [Java Application] C:\Users\nidhi\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.6.v20230204-1729\jre\bin\javaw.exe (Mar 29,
Output is: [1, 2, 3, 4]
Output is: [5, 6, 7, 8]
```

Practical No: 2

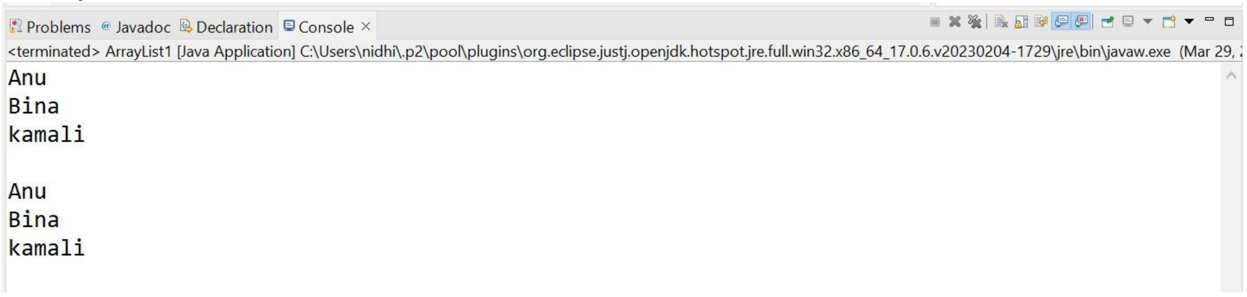
Assignments on List Interface

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

Code:

```java
package Mappack;
import java.util.ArrayList;
import java.util.List;
public class ArrayList1 {
public static void main(String args[]) {
List<String> str = new ArrayList<String>();
str.add("Anu");
str.add("Bina");
str.add("kamali");
//Using the get method and for loop
for(int i=0;i<str.size();i++) {
System.out.println(str.get(i)+" ");
}
System.out.println();
//using the for each loop
for(String name: str) {
System.out.println(name +" ");
}
}
}
```

Output:



```
<terminated> ArrayList1 [Java Application] C:\Users\nidhi\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.6.v20230204-1729\jre\bin\javaw.exe  (Mar 29,
Anu
Bina
kamali

Anu
Bina
kamali
```

Practical No: 3

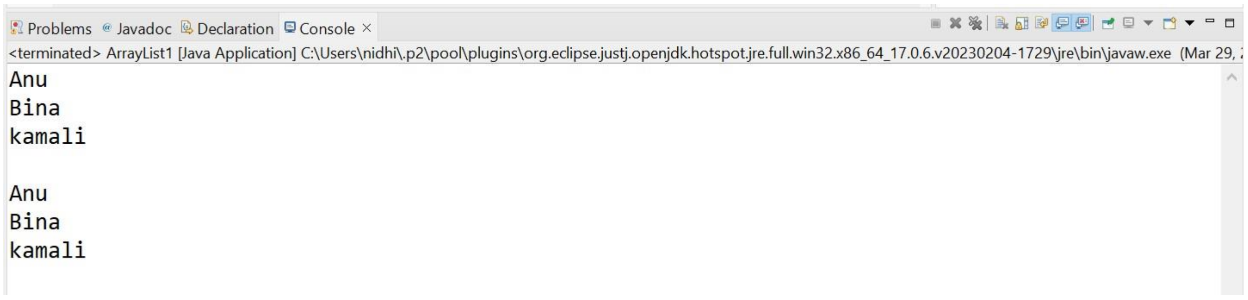
Assignments on Set Interface

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

Code:

```
package Mappack;
import java.util.Arrays;
import java.util.HashSet;
import java.util.Set;
public class SetOperations {
public static void main(String args[])
{
Integer[] A = {22, 45,33, 66, 55, 34, 77};
Integer[] B = {33, 2, 83, 45, 3, 12, 55};
Set<Integer> set1 = new HashSet<Integer>();
set1.addAll(Arrays.asList(A)); Set<Integer> set2 =
new HashSet<Integer>();
set2.addAll(Arrays.asList(B));
// Finding Union of set1 and set2
Set<Integer> union_data = new HashSet<Integer>(set1);
union_data.addAll(set2);
System.out.print("Union of set1 and set2 is:");
System.out.println(union_data);
// Finding Intersection of set1 and set2
Set<Integer> intersection_data = new HashSet<Integer>(set1);
intersection_data.retainAll(set2);
System.out.print("Intersection of set1 and set2 is:");
System.out.println(intersection_data);
// Finding Difference of set1 and set2
Set<Integer> difference_data = new HashSet<Integer>(set1);
difference_data.removeAll(set2);
System.out.print("Difference of set1 and set2 is:");
System.out.println(difference_data);
}
}
```

Output:



```
<terminated> ArrayList1 [Java Application] C:\Users\nidhi\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.6.v20230204-1729\jre\bin\javaw.exe  (Mar 29, 
Anu
Bina
kamali

Anu
Bina
kamali
```

Practical No: 4
Assignments on Map Interface

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

a. Items in the map.

b. Remove items from the map

c. Search specific key from the map

d. Get value of the specified key

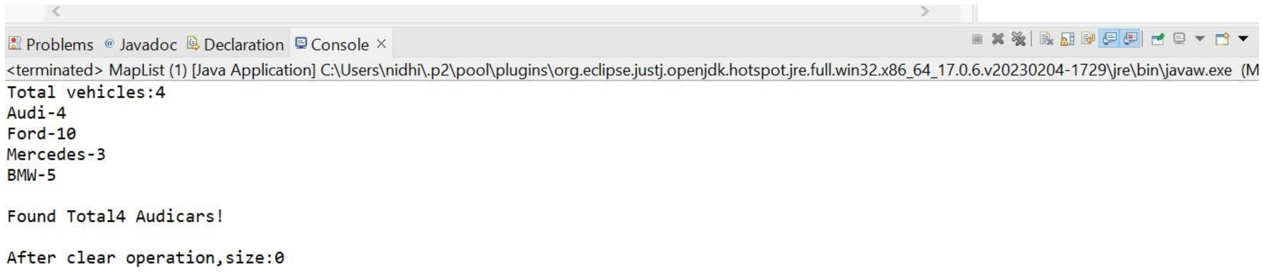
e. Insert map elements of one map in to other map.

f. Print all keys and values of the map.

Code:

```
package Mappack;
import java.util.HashMap;import
java.util.Map;
public class MapList {
public static void main(String[] args) {
// TODO Auto-generated method stub
Map<String,Integer> vehicles=new HashMap<>();
//Add some vehicles
vehicles.put("BMW", 5);
vehicles.put("Mercedes", 3);
vehicles.put("Audi", 4);
vehicles.put("Ford", 10);
System.out.println("Total vehicles:"+vehicles.size());
//Iterate over all vehicles,using the Keyset method.
for(String key:vehicles.keySet()) System.out.println(key+"-
"+vehicles.get(key));System.out.println();
String searchkey="Audi";
if(vehicles.containsKey(searchkey))
System.out.println("Found Total"+vehicles.get(searchkey)+" "+searchkey+"cars!\n");
//Clear all the values
vehicles.clear();
//Equal to zero
System.out.println("After clear operation,size:"+vehicles.size());
}
}
```

Output:



```
<terminated> MapList (1) [Java Application] C:\Users\nidhi\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.6.v20230204-1729\jre\bin\javaw.exe (M
Total vehicles:4
Audi-4
Ford-10
Mercedes-3
BMW-5

Found Total4 Audicars!

After clear operation,size:0
```

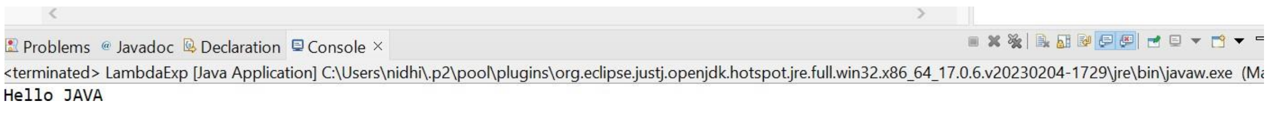
Practical No.5

Assignments on Lambda Expression

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

```
package Mappack;
interface MyFunctionalInterface
{
	public void say(String str1,String str2);
}
public class LambdaExp
{
	public static void main(String[] args)
	{
		// TODO Auto-generated method stub
		MyFunctionalInterface msg=(str1,str2) ->
					{
					System.out.println(str1+" "+str2);
					};
		msg.say("Hello", "JAVA");
	}
}
```

Output:



```
<terminated> LambdaExp [Java Application] C:\Users\nidhi\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.6.v20230204-1729\jre\bin\javaw.exe (M
Hello JAVA
```

Practical No.6
Assignments based on web application development using JSP

1. Write a JSP page to display the Registration form (Make your own assumptions)

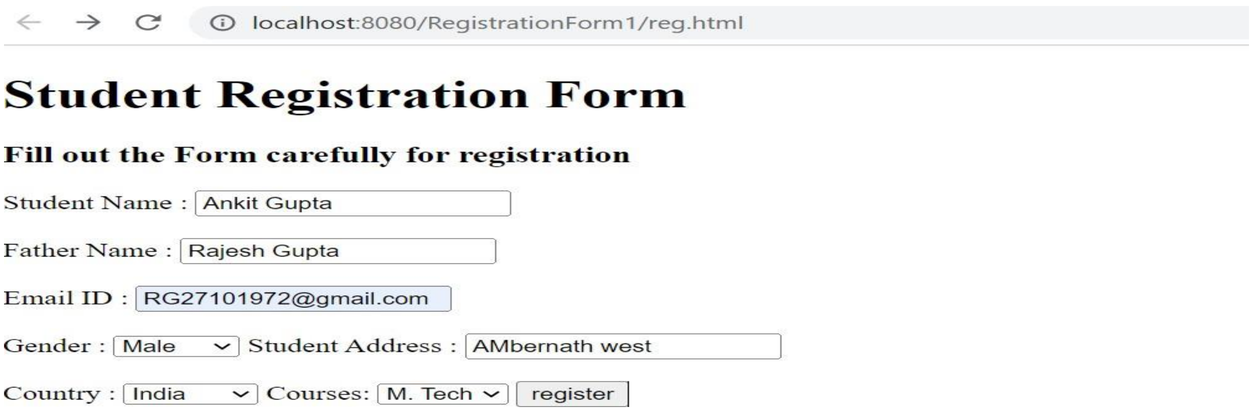
Reg.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Student Registration Form</title>
</head>
<body>
      <h1>Student Registration Form</h1>
      <h3>Fill out the Form carefully for registration</h3>
      <form action="register.jsp">
            Student Name: <input type="text" name="name"><br/><br/>
            Father Name: <input type="text" name="fname"><br/><br/>
            EmailID: <input type="text" name="email"><br/><br/>
            Gender: <select name="gender">
                          <option>Male</option>
                          <option>FeMale</option>
                          <option>Other</option>
                    </select>
            Student Address: <input type="text" name="address"><br/><br/>
            Country: <select name="country">
                          <option>India</option>
                          <option>China</option>
                          <option>America</option>
                          <option>Other</option>
                     </select>
            Courses:
                    <select name="Course">
                          <option>B. Tech</option>
                          <option>M. Tech</option>
                          <option>MBA</option>
                          <option>Other</option>
                    </select>
            <input type="submit"  value="register">
      </form>
</body>
</html>
```

Register.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Student Registration      Form</title>
</head>
<body>
<%
String name =request.getParameter("name");
String fname =request.getParameter("fname");
String emailid =request.getParameter("email");
String gender =request.getParameter("gender");
String address =request.getParameter("address");
String Country =request.getParameter("country");
String course =request.getParameter("course");
out.print("Name :" +name+"<br/>");
out.print("Father Name :" +fname+"<br/>");
out.print("Email Address :" +emailid+"<br/>");
out.print("Gender :" +gender+"<br/>");
out.print("Address :" +address+"<br/>");
out.print("Country :" +Country+"<br/>");
out.print("Course :" +course+"<br/>");
%>
</body>
</html>
```

Output:


localhost:8080/RegistrationForm1/reg.html

Student Registration Form

Fill out the Form carefully for registration

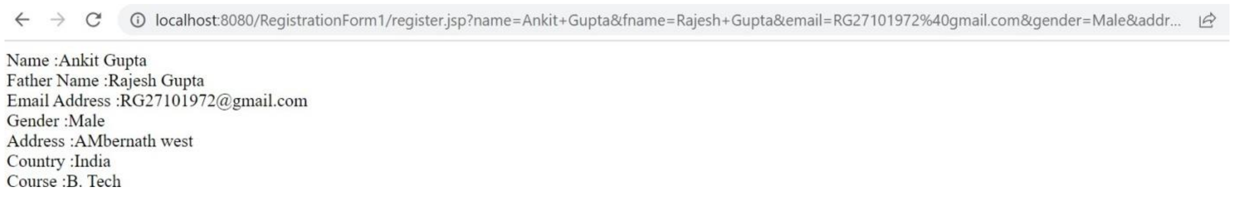
Student Name : Ankit Gupta

Father Name : Rajesh Gupta

Email ID : RG27101972@gmail.com

Gender : Male Student Address : AMbernath west

Country : India Courses: M. Tech register


localhost:8080/RegistrationForm1/register.jsp?name=Ankit+Gupta&fname=Rajesh+Gupta&email=RG27101972%40gmail.com&gender=Male&addr...

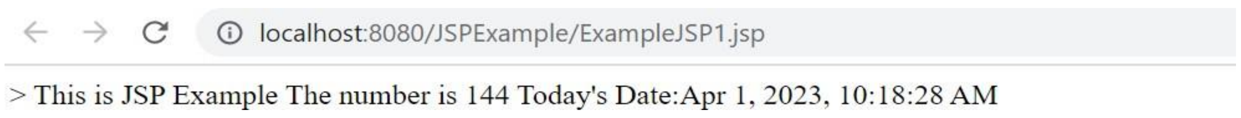
```
Name :Ankit Gupta
Father Name :Rajesh Gupta
Email Address :RG27101972@gmail.com
Gender :Male
Address :AMbernath west
Country :India
Course :B. Tech
```

2. Write a JSP program that demonstrates the use of JSP declaration, scriptlet, directives,expression.

ExampleJSP1.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>JSP Example</title>
</head>
<body>
<%--This is a JSP Example with scriptlets,comments, expressions --%>>
<%out.println("This is JSP Example"); %>
<%out.println("The number is"); %>
<%! int num12=12;int num32=12; %>
<%=num12*num32 %>
Today's Date:<%=(new java.util.Date().toLocaleString()) %>
</body>
</html>
```

Output:



localhost:8080/JSPExample/ExampleJSP1.jsp

```
> This is JSP Example The number is 144 Today's Date:Apr 1, 2023, 10:18:28 AM
```

PRACTICAL NO.7

Assignment based Spring Framework

1. Write a program to print “Hello World” using spring framework

pom.xml

```xml
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>springcore_example</groupId>
  <artifactId>springcore_example</artifactId>
  <version>0.0.1-SNAPSHOT</version>

  <dependencies>
        <dependency>
              <groupId>org.springframework</groupId>
              <artifactId>spring-core</artifactId>
              <version>4.0.0.RELEASE</version>
        </dependency>
        <dependency>
              <groupId>org.springframework</groupId>
              <artifactId>spring-context</artifactId>
              <version>4.0.0.RELEASE</version>
        </dependency>
  </dependencies>
</project>
```

HELLOBean.java

```java
package springcore_example;
public class HELLOBean
{
      private String name;
      public String getEmployeeName()
      {
            return name;
      }
      public void setName(String name)
      {
            this.name=name;
      }
      public void SayHello()
      {
            System.out.println("Hello Spring Framework example"+this.name);
      }
}
```

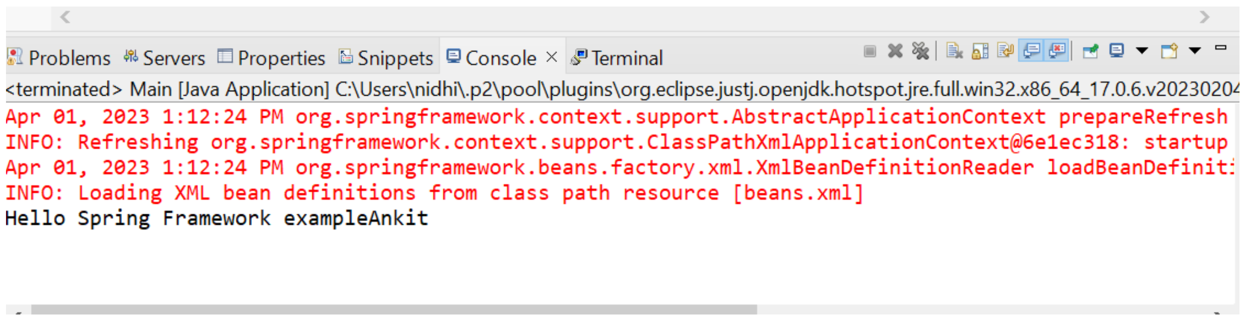
Main.java

```
package springcore_example;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
public class Main {
	private static ApplicationContext context;
	public static void main(String[] args) {
		// TODO Auto-generated method stub
		context=new ClassPathXmlApplicationContext("beans.xml");
		HELLOBean hlobean=(HELLOBean)context.getBean("Hellobean");
		hlobean.SayHello();
	}
}
```

Beans.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
To change this license header, choose License Headers in Project Properties.
To change this template file, choose Tools | Templates
and open the template in the editor.
-->
<beans xmlns = "http://www.springframework.org/schema/beans" xmlns:xsi =
"http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation =
"http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-3.0.xsd">
<bean id="Hellobean" class="springcore_example.HELLOBean">
<property name="name" value="Ankit"></property>
</bean>
</beans>
```

OUTPUT:



```
<terminated> Main [Java Application] C:\Users\nidhi\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.6.v2023020
Apr 01, 2023 1:12:24 PM org.springframework.context.support.AbstractApplicationContext prepareRefresh
INFO: Refreshing org.springframework.context.support.ClassPathXmlApplicationContext@6e1ec318: startup
Apr 01, 2023 1:12:24 PM org.springframework.beans.factory.xml.XmlBeanDefinitionReader loadBeanDefinit
INFO: Loading XML bean definitions from class path resource [beans.xml]
Hello Spring Framework exampleAnkit
```

PRACTICAL NO . 9
Assignment based Spring JDBC

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

1.Create Class Student :

```
package com.jdbctemplate;
public class Student {
            private Integer age;
            private String name;
            private Integer id;
            public void setAge(Integer age)
             {
               this.age = age;
            }
            public Integer getAge()
             {
               return age;
            }
            public void setName(String name)
            {
               this.name = name;
            }
            public String getName()
             {
                  return name;
            }
            public void setId(Integer id)
             {
               this.id = id;
            }
            public Integer getId()
             {
               return id;
            }
}
```

2.Create Class StudentMapper

```
package com.jdbctemplate;
import java.sql.ResultSet;
import java.sql.SQLException;
import org.springframework.jdbc.core.RowMapper;
public class StudentMapper implements RowMapper {

        public Student mapRow(ResultSet rs, int rowNum) throws SQLException {
                Student student = new Student();
                student.setId(rs.getInt("id"));
                student.setName(rs.getString("name"));
                student.setAge(rs.getInt("age"));
                return student;
              }
            }
```

3.Create Class

StudentDAO

```
package com.jdbctemplate;
import java.util.List;
import javax.sql.DataSource;
public interface StudentDAO {
   /**
   * This is the method to be used to initialize
   * database resources ie. connection.
   */
 public void setDataSource(DataSource ds);
 /**
   * This is the method to be used to create
   * a record in the Student table.
 */
 public void create(String name, Integer age);
 /**
   * This is the method to be used to list down
   * a record from the Student table corresponding
   * to a passed student id.
 */
 public Student getStudent(Integer id);
 /**
   * This is the method to be used to list down
   * all the records from the Student table.
 */
 public List<Student> listStudents();
 /**
   * This is the method to be used to delete
   * a record from the Student table corresponding
   * to a passed student id.
 */
 public void delete(Integer id);
 /**
   * This is the method to be used to update
   * a record into the Student table.
 */
 public void update(Integer id, Integer age);
}
```

4. Create Class StudentJDBCTemplate

```
package com.jdbctemplate;
import java.util.List;
import javax.sql.DataSource;
import org.springframework.jdbc.core.JdbcTemplate;
public class StudentJDBCTemplate implements StudentDAO {
            private DataSource dataSource;
            private JdbcTemplate jdbcTemplateObject;
            public void setDataSource(DataSource dataSource) {
                        this.dataSource = dataSource;
                        this.jdbcTemplateObject = new JdbcTemplate(dataSource);
                     }
                     public void create(String name, Integer age) {
                       String SQL = "insert into Student (name, age) values (?,? )";
                       jdbcTemplateObject.update( SQL,new Object[]{name, age});
                       System.out.println("Created Record Name = " + name + " Age = " + age);
                       return;
                     }
                     public Student getStudent(Integer id) {
                       String SQL = "select * from Student where id = ?";
                       Student student = (Student) jdbcTemplateObject.queryForObject(SQL,
                         new Object[]{id}, new StudentMapper());
                       return student;
                     }
                     public List<Student> listStudents() {
                       String SQL = "select * from Student";
                       List <Student> students = jdbcTemplateObject.query(SQL, new StudentMapper());
                       return students;
                     }
                     public void delete(Integer id) {
                       String SQL = "delete from Student where id = ?";
                       jdbcTemplateObject.update(SQL,new Object[]{id});
                       System.out.println("Deleted Record with ID = " + id );
                       return;
                     }
                     public void update(Integer id, Integer age){
                       String SQL = "update Student set age = ? where id = ?";
                       jdbcTemplateObject.update(SQL,new Object[]{age, id});
                       System.out.println("Updated Record with ID = " + id );
```

```
                              return;
                             }
 }
```

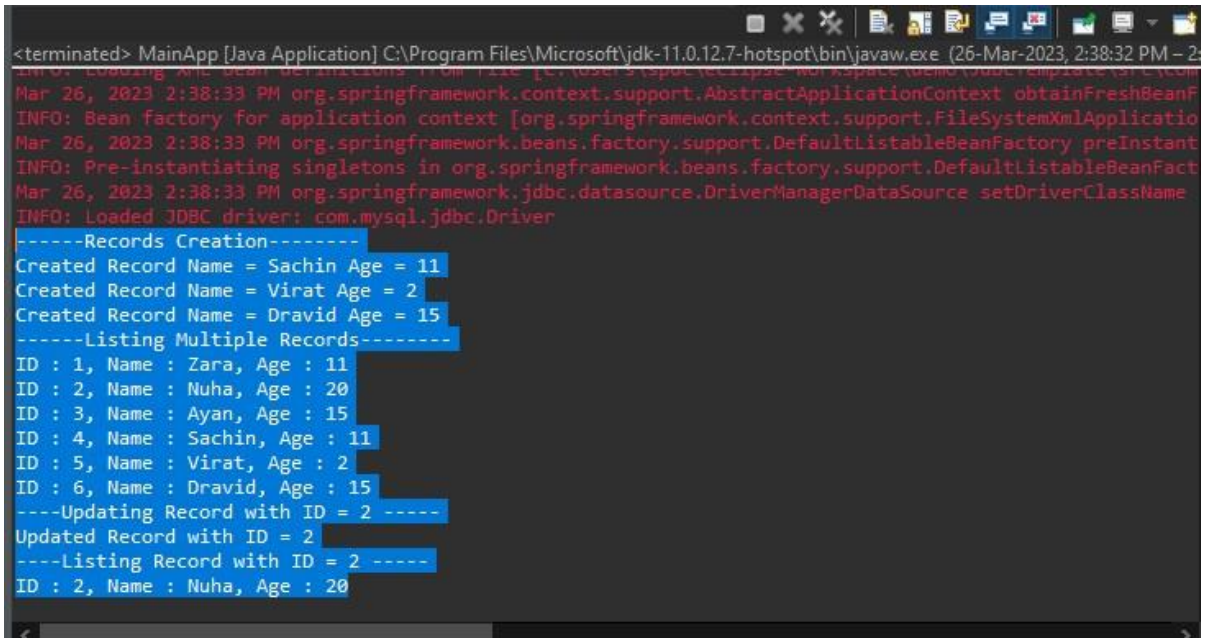
5.Create Class MainApp

```
package com.jdbctemplate;
import java.util.List;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.FileSystemXmlApplicationContext;
 public class MainApp {
                  public static void main(String[] args) {
                   //pplicationContext context = new
 ClassPathXmlApplicationContext("C:\\Users\\spdc\\eclipse-
 workspace\\demo\\JdbcTemplate\\src\\com\\jdbctemplate\\Beans.xml");
 ApplicationContext context = new
 FileSystemXmlApplicationContext("C:\\Users\\spdc\\eclipse-
 workspace\\demo\\JdbcTemplate\\src\\com\\jdbctemplate\\Beans.xml");
 StudentJDBCTemplate studentJDBCTemplate =
 (StudentJDBCTemplate)context.getBean("studentJDBCTemplate");
                    System.out.println("------Records Creation ------- " );
                    studentJDBCTemplate.create("Sachin", 11);
                    studentJDBCTemplate.create("Virat", 2);
                    studentJDBCTemplate.create("Dravid", 15);
                    System.out.println("------Listing Multiple Records ------- " );
                    List<Student> students = studentJDBCTemplate.listStudents();
                    for (Student record : students) {
                       System.out.print("ID : " + record.getId() );
                       System.out.print(", Name : " + record.getName() );
                       System.out.println(", Age : " + record.getAge());
                    }
                    System.out.println("----Updating Record with ID = 2 ----- " );
                    studentJDBCTemplate.update(2, 20);
                    System.out.println("----Listing Record with ID = 2------" );
                    Student student = studentJDBCTemplate.getStudent(2);
                    System.out.print("ID : " + student.getId() );
                    System.out.print(", Name : " + student.getName() );
                    System.out.println(", Age : " + student.getAge());
                 }
 }
```

6.Create Beans.xml

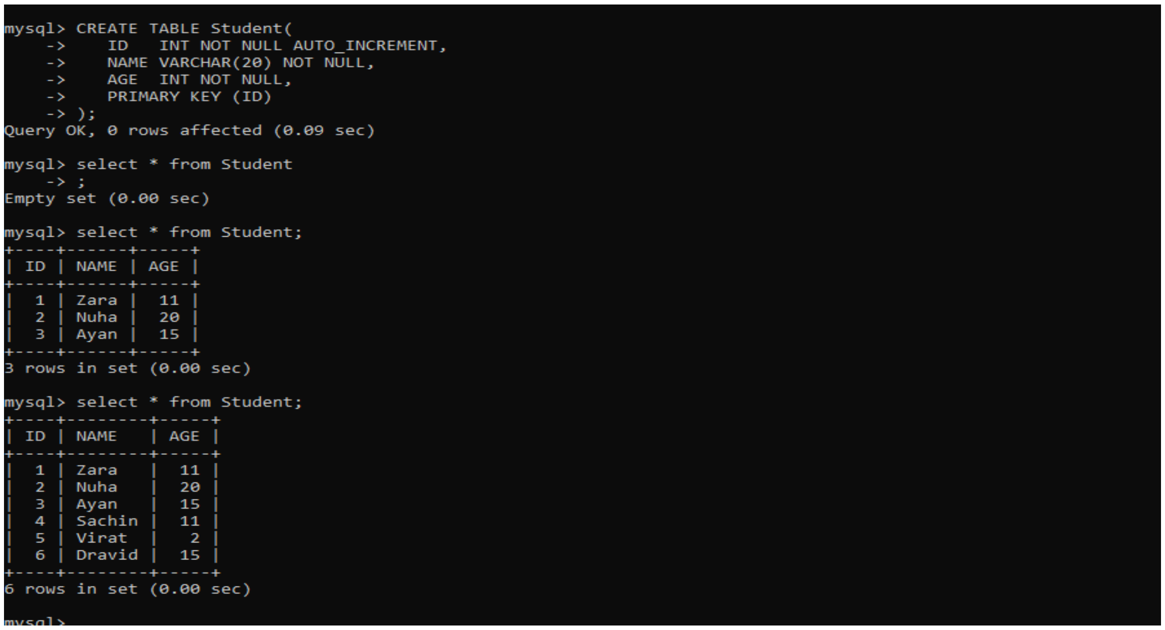
```xml
<?xml version = "1.0" encoding = "UTF-8"?>
<beans xmlns = "http://www.springframework.org/schema/beans"
  xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation = "http://www.springframework.org/schema/beans
  http://www.springframework.org/schema/beans/spring-beans-3.0.xsd ">
  <!-- Initialization for data source -->
  <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/test"/>
    <property name = "username" value = "root"/>
    <property name = "password" value = "12345"/>
  </bean>
  <!-- Definition for studentJDBCTemplate bean -->
  <bean id = "studentJDBCTemplate"
    class = "com.jdbctemplate.StudentJDBCTemplate">
    <property name = "dataSource" ref = "dataSource" />
  </bean>
</beans>
```

Output in eclipse :



```
<terminated> MainApp [Java Application] C:\Program Files\Microsoft\jdk-11.0.12.7-hotspot\bin\javaw.exe  (26-Mar-2023, 2:38:32 PM – 2
INFO: Loading XML bean definitions from file [C:\Users\spoc\eclipse-workspace\demo\JdbcTemplate\src\com
Mar 26, 2023 2:38:33 PM org.springframework.context.support.AbstractApplicationContext obtainFreshBeanF
INFO: Bean factory for application context [org.springframework.context.support.FileSystemXmlApplicatio
Mar 26, 2023 2:38:33 PM org.springframework.beans.factory.support.DefaultListableBeanFactory preInstant
INFO: Pre-instantiating singletons in org.springframework.beans.factory.support.DefaultListableBeanFact
Mar 26, 2023 2:38:33 PM org.springframework.jdbc.datasource.DriverManagerDataSource setDriverClassName
INFO: Loaded JDBC driver: com.mysql.jdbc.Driver
------Records Creation--------
Created Record Name = Sachin Age = 11
Created Record Name = Virat Age = 2
Created Record Name = Dravid Age = 15
------Listing Multiple Records--------
ID : 1, Name : Zara, Age : 11
ID : 2, Name : Nuha, Age : 20
ID : 3, Name : Ayan, Age : 15
ID : 4, Name : Sachin, Age : 11
ID : 5, Name : Virat, Age : 2
ID : 6, Name : Dravid, Age : 15
----Updating Record with ID = 2 -----
Updated Record with ID = 2
----Listing Record with ID = 2 -----
ID : 2, Name : Nuha, Age : 20
```

Output in Mysql:



```
mysql> CREATE TABLE Student(
    ->    ID   INT NOT NULL AUTO_INCREMENT,
    ->    NAME VARCHAR(20) NOT NULL,
    ->    AGE  INT NOT NULL,
    ->    PRIMARY KEY (ID)
    -> );
Query OK, 0 rows affected (0.09 sec)

mysql> select * from Student
    -> ;
Empty set (0.00 sec)

mysql> select * from Student;
+----+------+-----+
| ID | NAME | AGE |
+----+------+-----+
|  1 | Zara |  11 |
|  2 | Nuha |  20 |
|  3 | Ayan |  15 |
+----+------+-----+
3 rows in set (0.00 sec)

mysql> select * from Student;
+----+--------+-----+
| ID | NAME   | AGE |
+----+--------+-----+
|  1 | Zara   |  11 |
|  2 | Nuha   |  20 |
|  3 | Ayan   |  15 |
|  4 | Sachin |  11 |
|  5 | Virat  |   2 |
|  6 | Dravid |  15 |
+----+--------+-----+
6 rows in set (0.00 sec)

mysql>
```

PRACTICAL NO. 10

Assignment based Spring Boot and RESTful Web Services

1. Write a program to create a simple Spring Boot application that prints a message.

SpringexampleApplication.java

```
package Spring.springexample;
import org.springframework.boot.web.servlet.server.ServletWebServerFactory;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.FileSystemXmlApplicationContext;
@SpringBootApplication
public class SpringexampleApplication {
   public static void main(String[] args) {
          SpringApplication.run(SpringexampleApplication.class, args);
         ApplicationContext context = new
   FileSystemXmlApplicationContext("C:\\Users\\nidhi\\Downloads\\springexample
(1)\\springexample\\src\\main\\resources\\beans.xml");
          Sprinttest obj = (Sprinttest) context.getBean("helloWorld");
                       obj.getMessage();
   }

}
```

Sprinttest.java

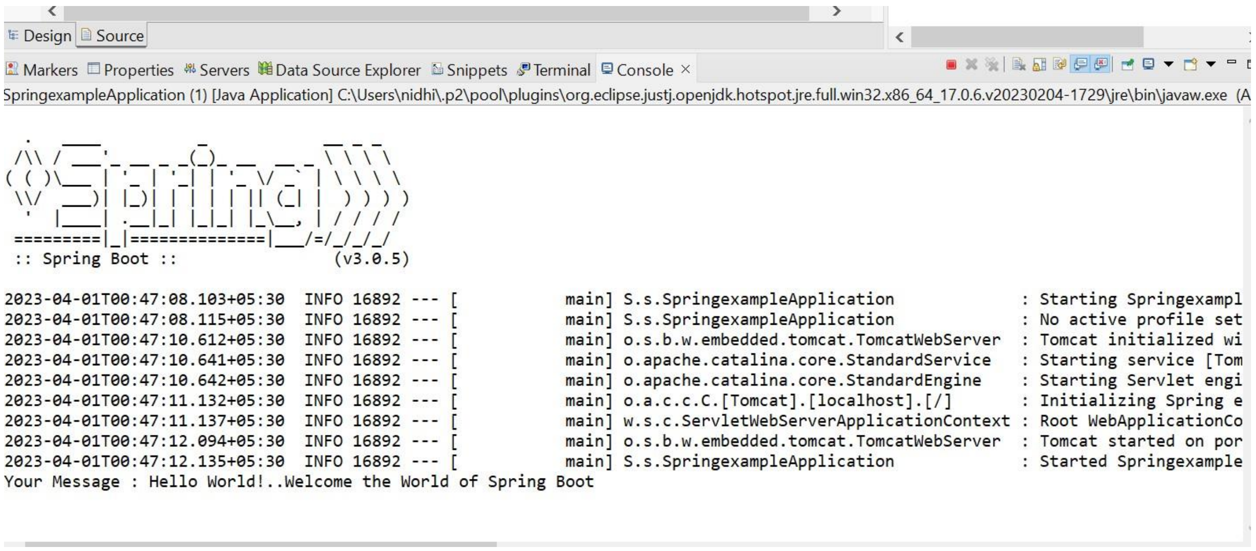
```
package Spring.springexample;
public class Sprinttest {
       private String message;
       public void setMessage(String message){
       this.message = message;
       }
       public void getMessage(){
       System.out.print("Your Message : " + message);
       }
}
```

Beans.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
To change this license header, choose License Headers in Project Properties.
To change this template file, choose Tools | Templates
and open the template in the editor.
-->

<beans xmlns = "http://www.springframework.org/schema/beans"
xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation = "http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-3.0.xsd">
<bean id = "helloWorld" class = "Spring.springexample.Sprinttest">
<property name = "message" value = "Hello World!..Welcome the World of Spring Boot"/>
</bean>
</beans>
```

Output:



SpringexampleApplication (1) [Java Application] C:\Users\nidhi\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.6.v20230204-1729\jre\bin\javaw.exe (A

```
  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::                (v3.0.5)

2023-04-01T00:47:08.103+05:30  INFO 16892 --- [           main] S.s.SpringexampleApplication             : Starting Springexampl
2023-04-01T00:47:08.115+05:30  INFO 16892 --- [           main] S.s.SpringexampleApplication             : No active profile set
2023-04-01T00:47:10.612+05:30  INFO 16892 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat initialized wi
2023-04-01T00:47:10.641+05:30  INFO 16892 --- [           main] o.apache.catalina.core.StandardService   : Starting service [Tom
2023-04-01T00:47:10.642+05:30  INFO 16892 --- [           main] o.apache.catalina.core.StandardEngine    : Starting Servlet engi
2023-04-01T00:47:11.132+05:30  INFO 16892 --- [           main] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring e
2023-04-01T00:47:11.137+05:30  INFO 16892 --- [           main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationCo
2023-04-01T00:47:12.094+05:30  INFO 16892 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on por
2023-04-01T00:47:12.135+05:30  INFO 16892 --- [           main] S.s.SpringexampleApplication             : Started Springexample
Your Message : Hello World!..Welcome the World of Spring Boot
```