Institute of Distance and Open Learning

Vidya Nagari, Kalina, Santacruz East – 400098.

CERTIFICATE

This is to certify that Mr. **YADAV MANISH RAJESH(40414)** of **Master in Computer Application** (MCA) Semester **I** has completed the specified term work in the subject of **ADVANCE JAVA** satisfactorily within this institute as laid down by University of Mumbai during the academic year **2024** to **2025**.

Subject In-charge	External Examiner	Coordinator – M.C.A

Institute of Distance and Open Learning (IDOL)

PCP CENTER: DTSS, MALAD

INDEX

Subject: <u>ADVANCE JAVA</u>

Sr. No.	Experiment Name	Date	Sign
1	Write a Java program to demonstrate the use of ArrayList		
2	Write a Java program to demonstrate the use of Vector		
3	Write a Java program to demonstrate the use of Stack		
4	Write a Java program to demonstrate the use of Java Map a.Java Map:Generic b.Java Map: comparing ByValues()		
5	Write a Java program to demonstrate the use of Lambda Expression a.Single Parameter b.Multiple Parameter		
6	JSP scriptlet tag that prints the user name		
7	JSP expression tag that prints current time		
8	JSP declaration tag that declares a field.		
9	JSP forward action tag with parameter.		
10	Simple example of JavaBean class		

Institute of Distance and Open Learning (IDOL)

PCP CENTER: DTSS, MALAD

INDEX

Subject: ADVANCE JAVA

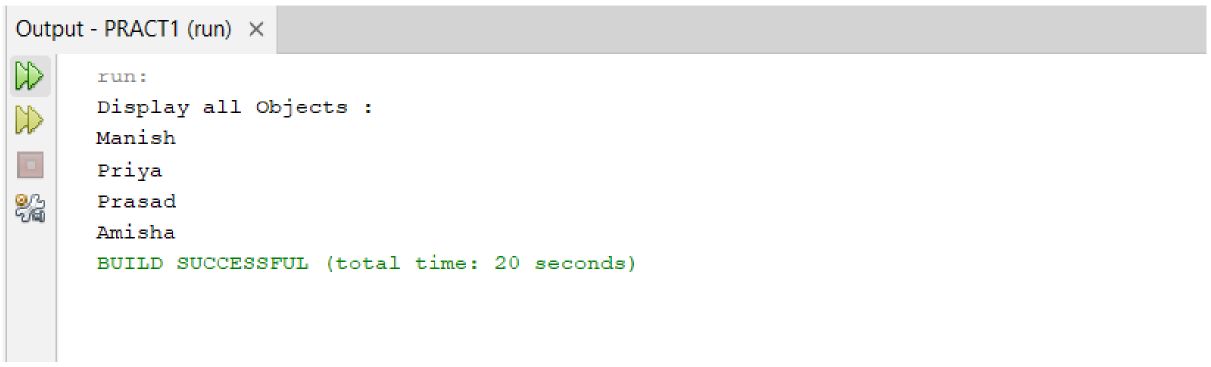
EXPERIMENT – I

<u>AIM:</u> Write a Java program to demonstrate the use of ArrayList

<u>CODE</u>:

```
package pract1;

import java.util.*;

public class PRACT1 {

    public static void main(String args[]) {

        ArrayList<String> list = new ArrayList<String>();//Creating arraylist

        list.add("Manish");//Adding object in arraylist

        list.add("Priya");

        list.add("Prasad");

        list.add("Amisha");

//Traversing list through Iterator

        Iterator itr = list.iterator();

        System.out.println("Display all Objects :");

        while (itr.hasNext()) {

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

        }

    }

}
```

<u>OUTPUT:</u>


Output - PRACT1 (run)

```
run:
Display all Objects :
Manish
Priya
Prasad
Amisha
BUILD SUCCESSFUL (total time: 20 seconds)
```

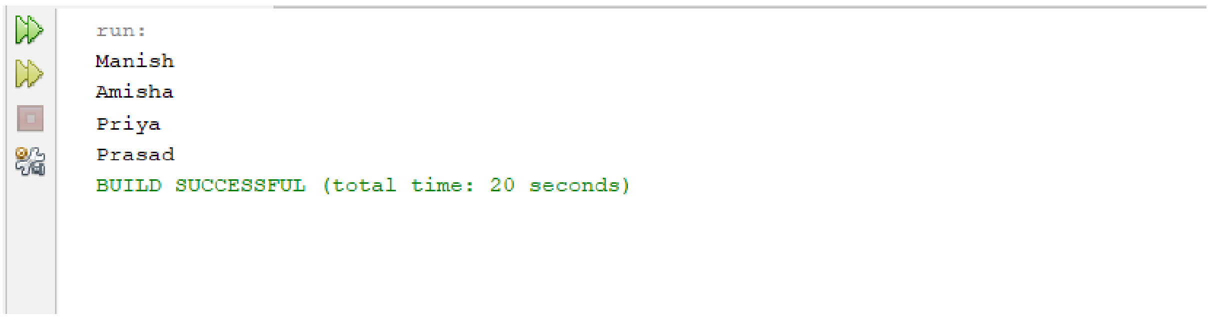
EXPERIMENT – II

AIM: Write a Java program to demonstrate the use of Vector

CODE:

```
import java.util.*;

public class Pract2

{

 public static void main(String args[])

{

Vector<String> v=new Vector<String>();

v.add("Manish");

v.add("Amisha");

v.add("Priya");

v.add("Prasad");

Iterator<String> itr=v.iterator();

while(itr.hasNext())

        {

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

        }

}

}
```

OUTPUT:



```
run:
Manish
Amisha
Priya
Prasad
BUILD SUCCESSFUL (total time: 20 seconds)
```

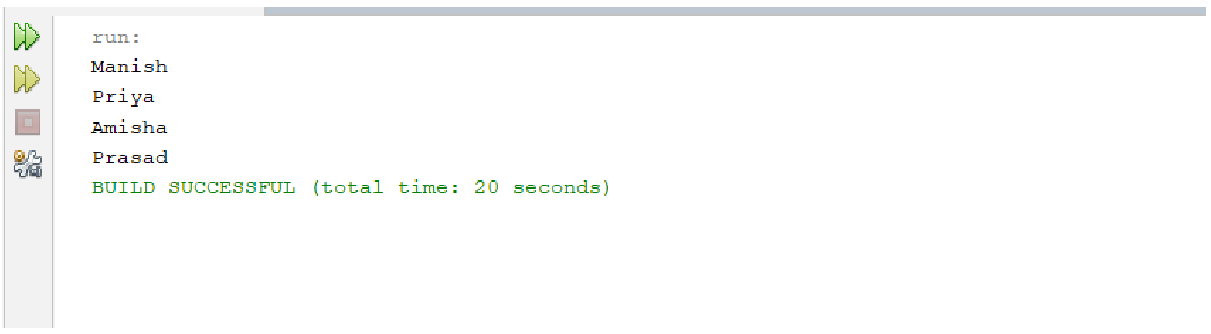
EXPERIMENT – III

AIM: Write a Java program to demonstrate the use of Stack

CODE:

```
import java.util.*;
public class Pract3 {
    public static void main(String args[]) {
        Stack<String> stack = new Stack<String>();
        stack.push("Manish");
        stack.push("Priya");
        stack.push("Amisha");
        stack.push("Prasad");
        stack.push("Garima");
        stack.pop();
        Iterator<String> itr = stack.iterator();
        while (itr.hasNext()) {
            System.out.println(itr.next());
        }
    }
}
```

OUTPUT:



```
run:
Manish
Priya
Amisha
Prasad
BUILD SUCCESSFUL (total time: 20 seconds)
```

EXPERIMENT – IV

AIM: Write a Java program to demonstrate the use of Java Map

a. Java Map:Generic

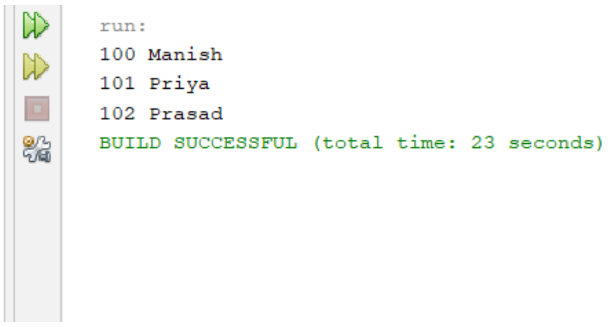
b. Java Map: comparing ByValues()

CODE:

a. Java Map:Generic

```
import java.util.*;
class Pract4{
public static void main(String args[])
  {
  Map<Integer,String> map=new HashMap<Integer,String>();
  map.put(100,"Manish");
  map.put(101,"Priya");
  map.put(102,"Prasad");
  //Elements can traverse in any order
for(Map.Entry m:map.entrySet())
{
   System.out.println(m.getKey()+" "+m.getValue());
  }
 }
}
```

OUTPUT:

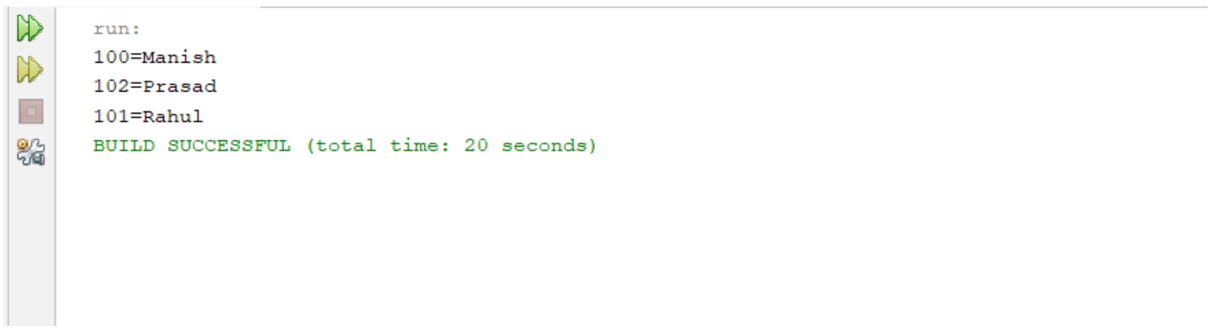


```
run:
100 Manish
101 Priya
102 Prasad
BUILD SUCCESSFUL (total time: 23 seconds)
```

b. Java Map: comparing ByValues()

```
import java.util.*;
class Pract4b {
public static void main(String args[]) {
        Map<Integer, String> map = new HashMap<Integer, String>();
        map.put(100, "Manish");
        map.put(101, "Rahul");
        map.put(102, "Prasad");
        //Returns a Set view of the mappings contained in this map
        map.entrySet()
                //Returns a sequential Stream with this collection as its source
                .stream()
                //Sorted according to the provided Comparator
                .sorted(Map.Entry.comparingByValue())
                //Performs an action for each element of this stream
                .forEach(System.out::println);
    }
}
```

OUTPUT:



```
run:
100=Manish
102=Prasad
101=Rahul
BUILD SUCCESSFUL (total time: 20 seconds)
```

EXPERIMENT – V

AIM: Write a Java program to demonstrate the use of Lambda Expression

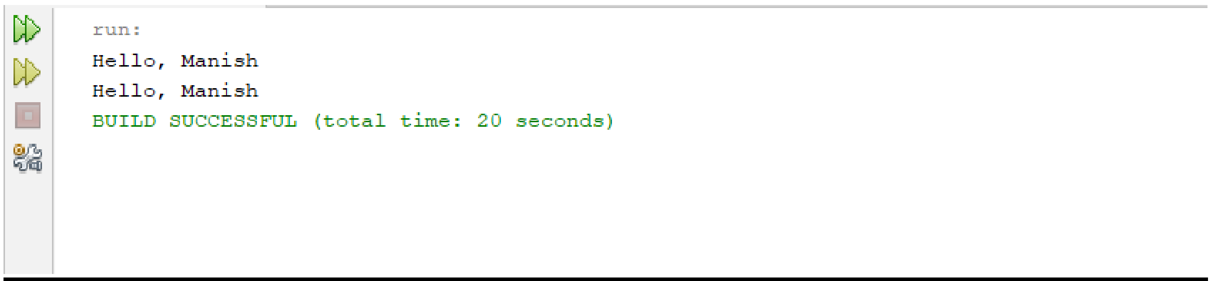
- **a. Single Parameter**
- **b. Multiple Parameter**

CODE:

a. Single Parameter

```
interface Sayable
{
public String say(String name);
}
public class Pract5a
{
public static void main(String[] args)
{
        // Lambda expression with single parameter.
        Sayable s1=(name)->{
return "Hello, "+name;
        };
        System.out.println(s1.say("Manish"));

        // You can omit function parentheses
        Sayable s2= name ->{
return "Hello, "+name;
        };
        System.out.println(s2.say("Manish"));
    }  }
```

OUTPUT:

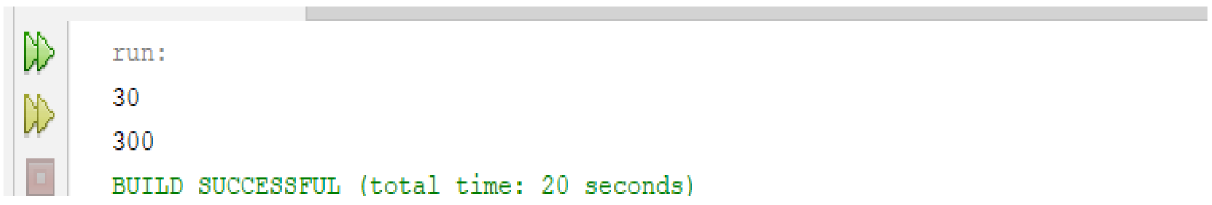


```
run:
Hello, Manish
Hello, Manish
BUILD SUCCESSFUL (total time: 20 seconds)
```

b. Multiple Parameter

```
interface Addable {

    int add(int a, int b);
}

public class Pract5b {

    public static void main(String[] args) {
        // Multiple parameters in lambda expression
        Addable ad1 = (a, b) -> (a + b);
        System.out.println(ad1.add(10, 20));

        // Multiple parameters with data type in lambda expression
        Addable ad2 = (int a, int b) -> (a + b);
        System.out.println(ad2.add(100, 200));
    }
}
```

OUTPUT:



```
run:
30
300
BUILD SUCCESSFUL (total time: 20 seconds)
```

EXPERIMENT – VI

AIM: JSP scriptlet tag that prints the user name

CODE:

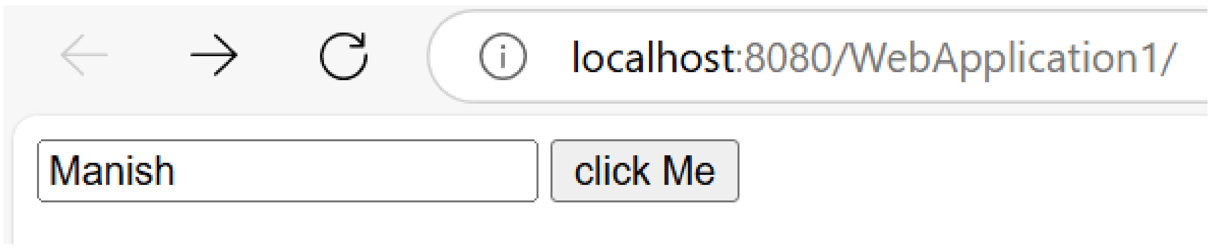
File: index.html

```
<html>
<body>
<form action="welcome.jsp">
<input type="text" name="username">
<input type="submit" value="click Me"><br/>
</form>
</body>
</html>
```

File: welcome.jsp

```
<html>
<body>
<%
String name=request.getParameter("username");
out.print("welcome "+username);
%>
</form>
</body>
</html>
```

OUTPUT:



EXPERIMENT – VII

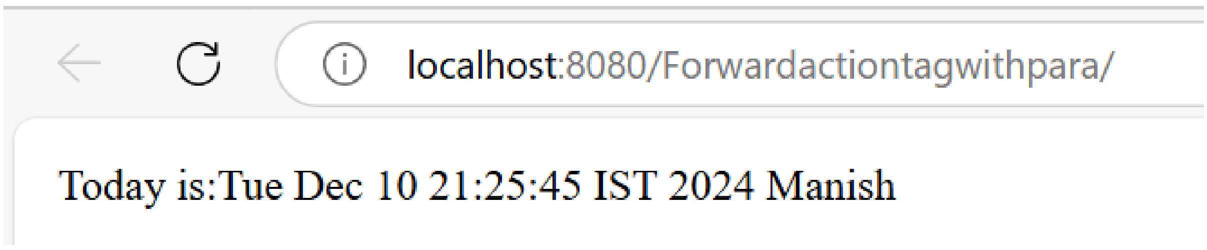
AIM: JSP expression tag that prints current time

CODE:

Index.jsp

```
<html>

<body>

Current Time: <%= java.util.Calendar.getInstance().getTime() %>

</body>

</html>
```

OUTPUT:



localhost:8080/Forwardactiontagwithpara/

```
Today is:Tue Dec 10 21:25:45 IST 2024 Manish
```

EXPERIMENT – VIII

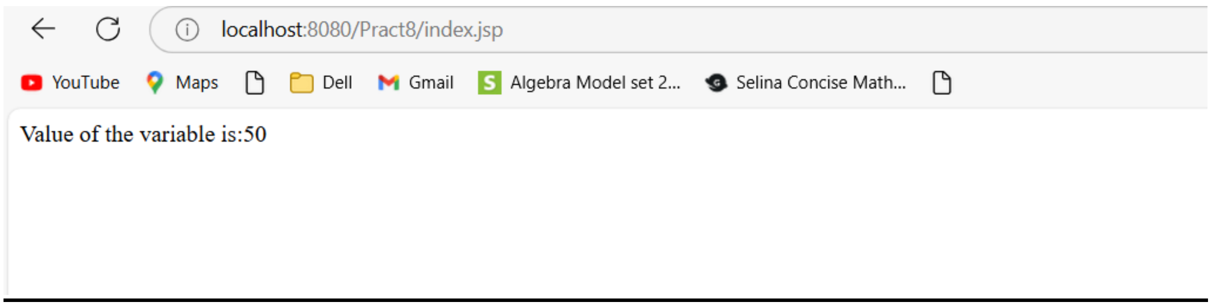
AIM: **JSP declaration tag that declares a field.**

CODE:

index.jsp

```
<html>
<body>
<%! int data=50; %>
<%= "Value of the variable is:"+data %>
</body>
</html>
```

OUTPUT:



```
Value of the variable is:50
```

EXPERIMENT – IX

AIM: **JSP forward action tag with parameter.**

CODE:

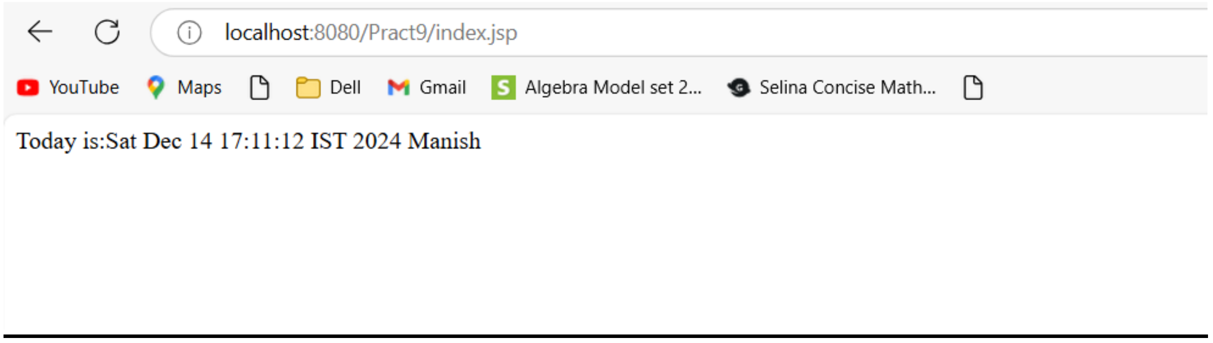
index.jsp

```
<html>
<body>
<h2> index page</h2>
<jsp:forward page="printdate.jsp" >
<jsp:param name="name" value="Manish" />
</jsp:forward>
</body>
</html>
```

printdate.jsp

```
<html>
<body>
<% out.print("Today is:"+java.util.Calendar.getInstance().getTime()); %>
<%= request.getParameter("name") %>
</body>
</html>
```

OUTPUT:



```
Today is:Sat Dec 14 17:11:12 IST 2024 Manish
```

EXPERIMENT – X

AIM: Simple example of JavaBean class

CODE:

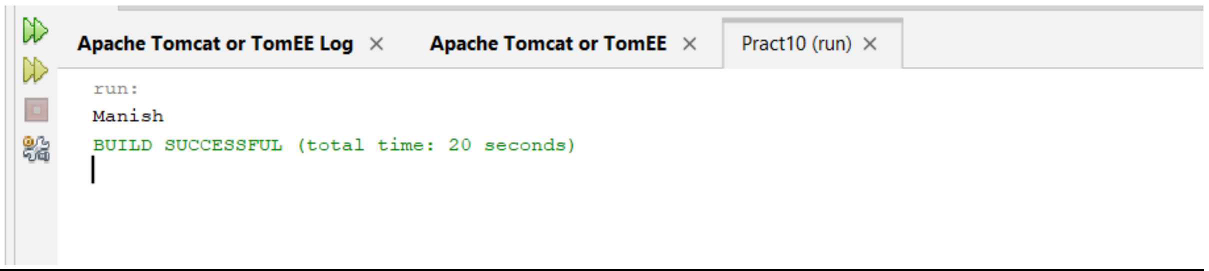
Student.java

```
public class Student implements java.io.Serializable
{
        private int id;
        private String name;
        public Student()
        {       }
               public void setId(int id)
                {
                this.id=id;
                }
              public int getId()
                {
                return id;
                }
               public void setName(String name)
                {
                this.name=name;
                }
              public String getName()
                {
                return name;
                }
}
```

Test.java

```
public class Test
{
public static void main(String args[])
{
Student s=new Student();//object is created
s.setName("Umesh");//setting value to the object
System.out.println(s.getName());
}
}
```

OUTPUT:



```
run:
Manish
BUILD SUCCESSFUL (total time: 20 seconds)
```

EXPERIMENT – XI

AIM: Demonstrate JSP Page Directive

a. Import

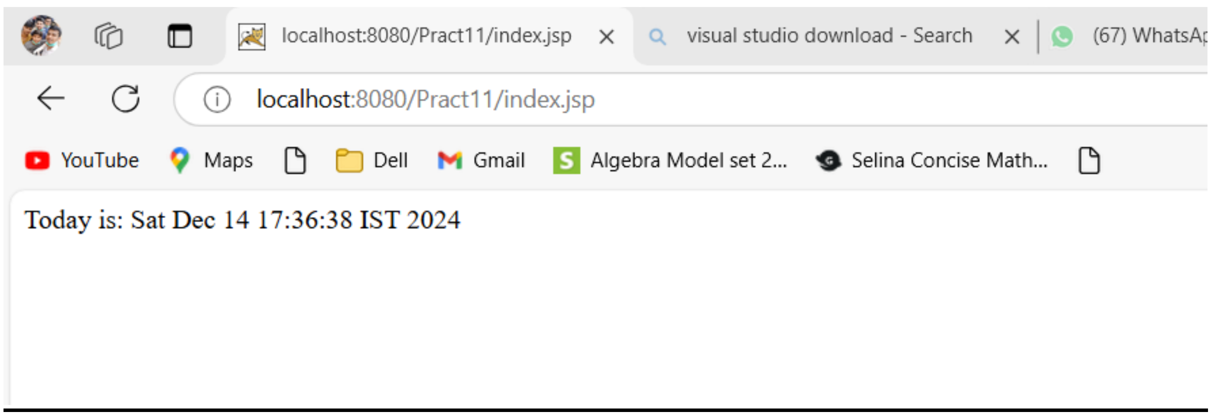
b. isErrorPage

CODE:

a. Import

```
<html>
<body>
<%@ page import="java.util.Date" %>
Today is: <%= new Date() %>
</body>
</html>
```

OUTPUT:

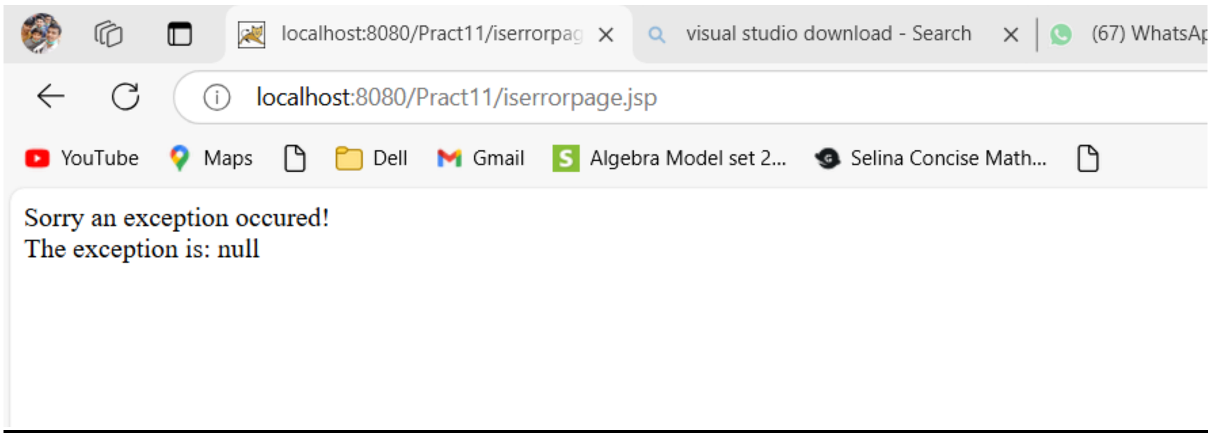

localhost:8080/Pract11/index.jsp
visual studio download - Search
(67) WhatsAp
localhost:8080/Pract11/index.jsp
YouTube
Maps
Dell
Gmail
Algebra Model set 2...
Selina Concise Math...

```
Today is: Sat Dec 14 17:36:38 IST 2024
```

b. isErrorPage

```
<html>
   <body>
      <%@ page isErrorPage="true" %>
      Sorry an exception occured!<br/>
      The exception is: <%= exception%>
   </body>
</html>
```

OUTPUT:


localhost:8080/Pract11/iserrorpag
visual studio download - Search
(67) WhatsAp
localhost:8080/Pract11/iserrorpage.jsp
YouTube
Maps
Dell
Gmail
Algebra Model set 2...
Selina Concise Math...

```
Sorry an exception occured!
The exception is: null
```

EXPERIMENT – XII

AIM: Demonstrate JSP Include Directive

CODE:

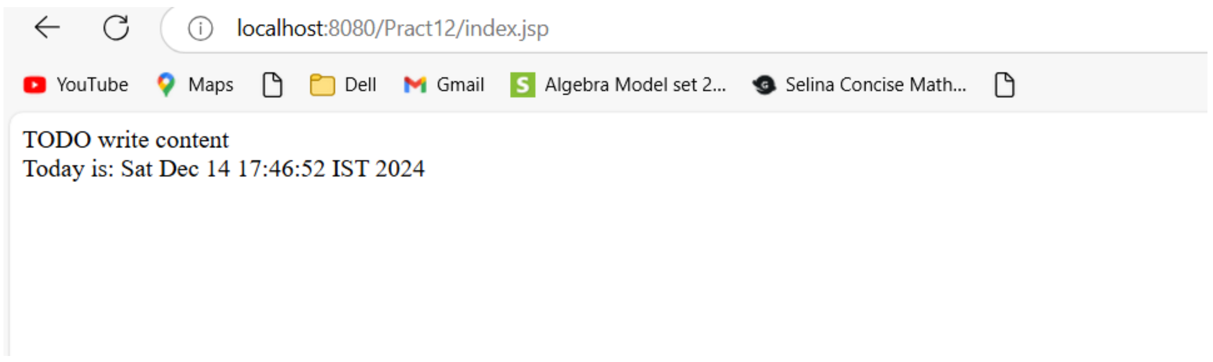
Index.html

```
<html>
   <head>
      <title>TODO supply a title</title>
      <meta charset="UTF-8">
      <meta name="viewport" content="width=device-width, initial-scale=1.0">
   </head>
   <body>
      <div>TODO write content</div>
   </body>
</html>
```

Index.jsp

```
<html>
<body>
<%@ include file="index.html" %>
Today is: <%= java.util.Calendar.getInstance().getTime() %>
</body>
</html>
```

OUTPUT:



```
TODO write content
Today is: Sat Dec 14 17:46:52 IST 2024
```

EXPERIMENT – XIII

AIM: Demonstrate JSP Tag Lib

CODE:

```
<html>
<body>
<%@ taglib uri="http://www.weburl.com/tags" prefix="mytag" %>
<mytag:currentDate/>
</body>
</html>
```

OUTPUT:

```
Current Date and Time: 2024-12-15 14:32:00
```

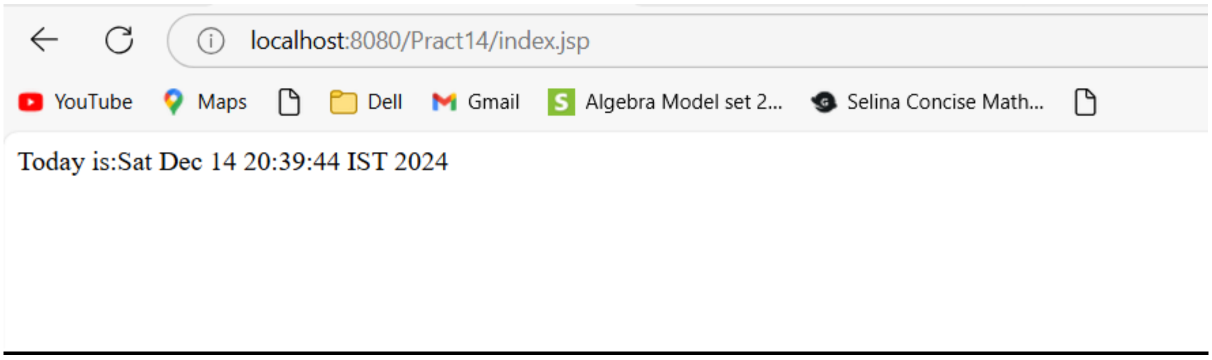
EXPERIMENT – XIV

AIM: Demonstrate JSP Implicit Object

CODE:

```
<html>

<body>

<% out.print("Today is:"+java.util.Calendar.getInstance().getTime()); %>

</body>

</html>
```

OUTPUT:



```
Today is:Sat Dec 14 20:39:44 IST 2024
```

EXPERIMENT – XV

AIM: Demonstrate Application Implicit Object

CODE:

Index.html

```
<html>
    <head>
        <title>TODO supply a title</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
    </head>
    <body>
        <form action="welcome">
            <input type="text" name="uname">
            <input type="submit" value="go"><br/>
        </form>
    </body>
</html>
```

welcome.jsp

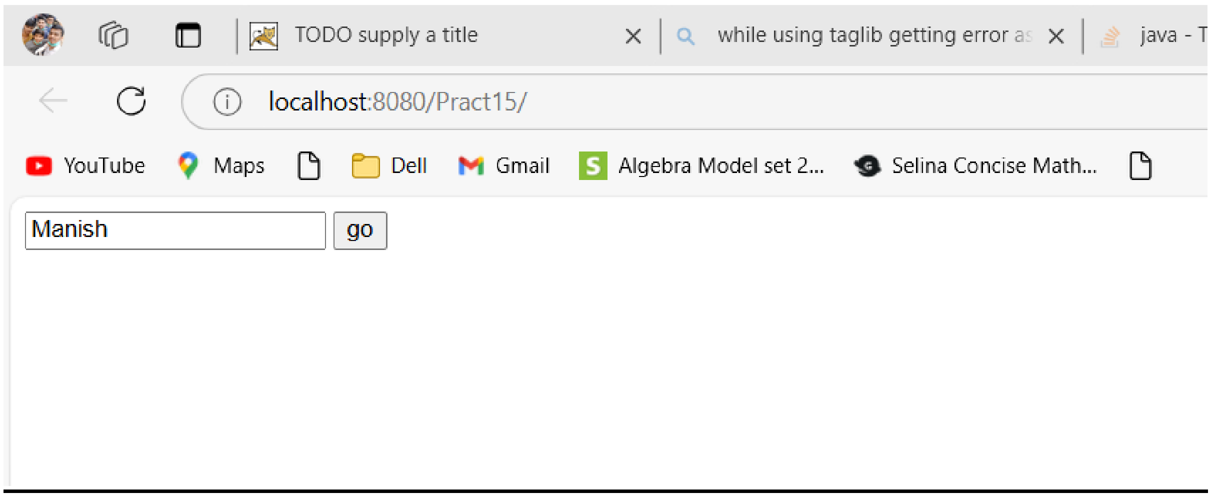
```
<%
out.print("Welcome "+request.getParameter("uname"));
String driver=application.getInitParameter("dname");
out.print("driver name is="+driver);
%>
```

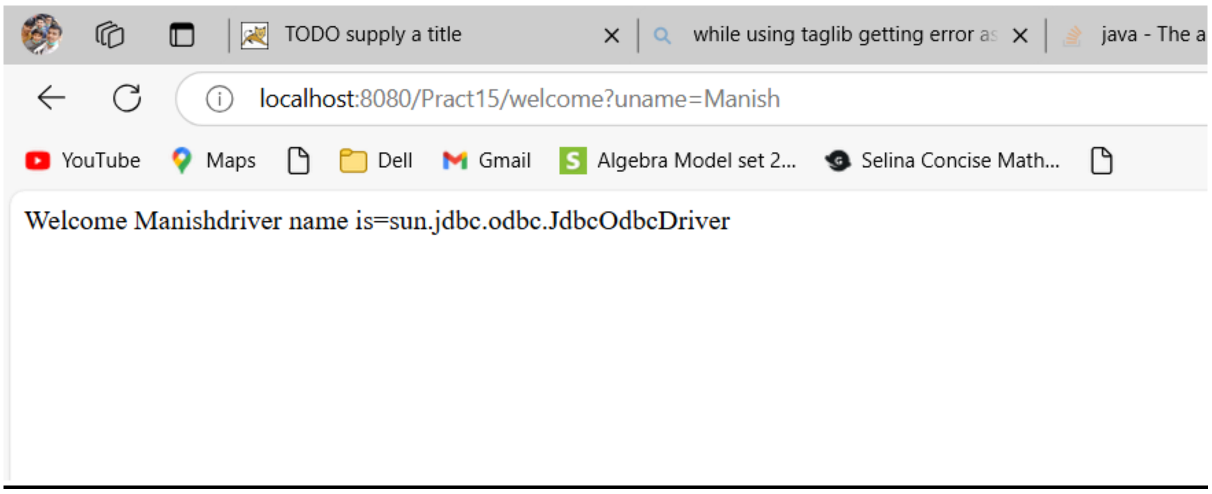
Web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app>
<servlet>
<servlet-name>Manish Yadav</servlet-name>
<jsp-file>/welcome.jsp</jsp-file>
</servlet>
```

```
<servlet-mapping>
<servlet-name>Manish Yadav</servlet-name>
<url-pattern>/welcome</url-pattern>
</servlet-mapping>

<context-param>
<param-name>dname</param-name>
<param-value>sun.jdbc.odbc.JdbcOdbcDriver</param-value>
</context-param>
</web-app>
```

OUTPUT:





```
Welcome Manishdriver name is=sun.jdbc.odbc.JdbcOdbcDriver
```

EXPERIMENT – XVI

AIM: Demonstrate Session Implicit Object

CODE:

Index.html

```
<html>
<body>
<form action="welcome.jsp">
<input type="text" name="uname">
<input type="submit" value="go"><br/>
</form>
</body>
</html>
```

Welcome.jsp

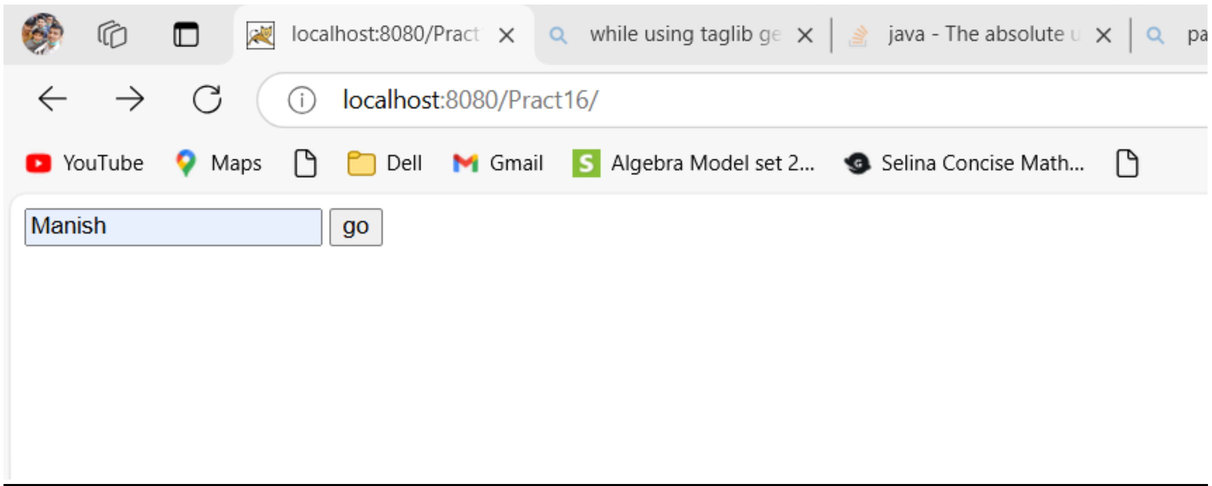
```
<html>
<body>
<%
String name=request.getParameter("uname");
out.print("Welcome "+name);
session.setAttribute("user",name);
%>
<a href="second.jsp">second jsp page</a>
</body>
</html>
```

Second.jsp

```
<html>
<body>
<%

String name=(String)session.getAttribute("user");
out.print("Hello "+name);
```

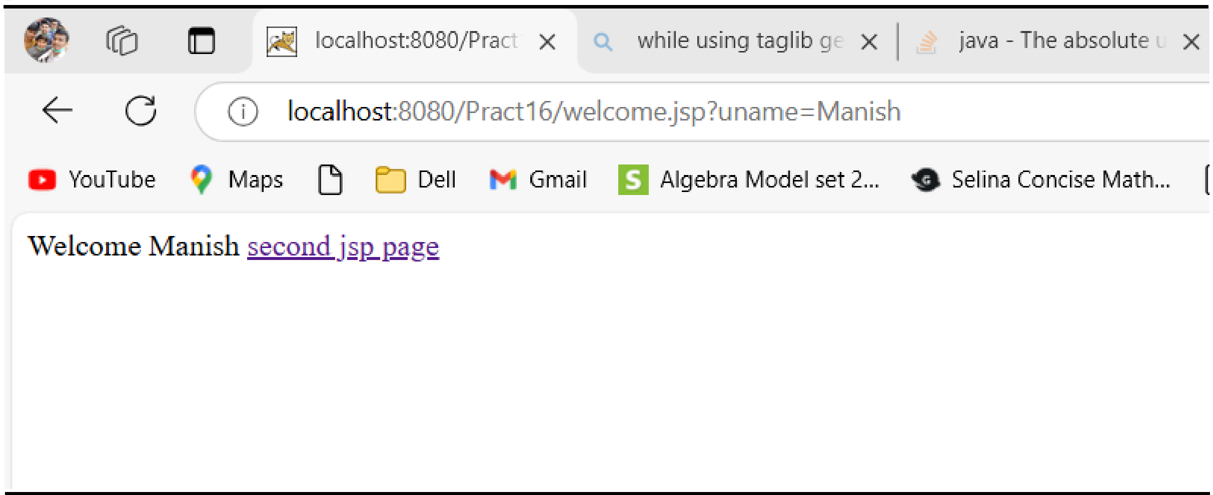
```
%>
</body>
</html>
```

OUTPUT:



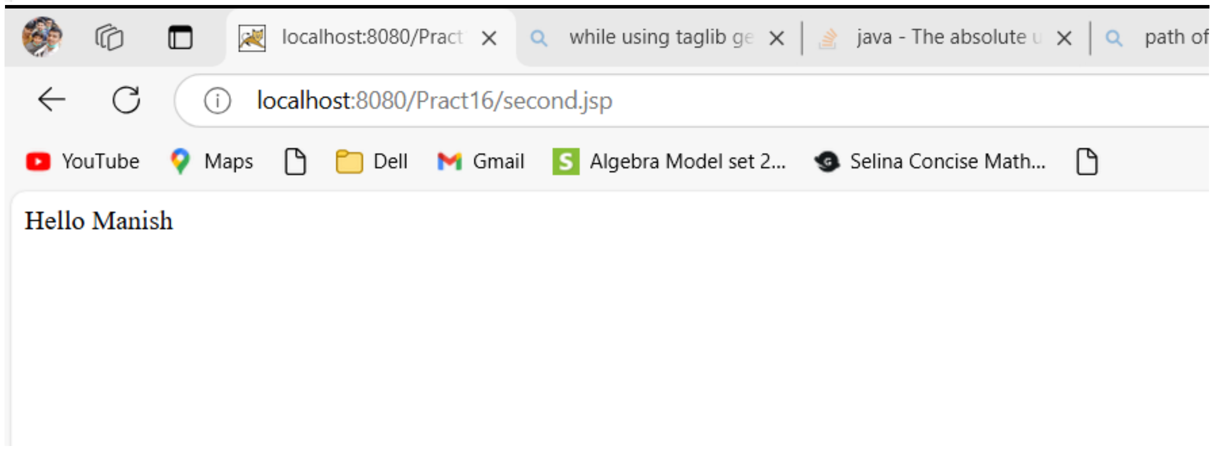
localhost:8080/Pract16/

Manish go



localhost:8080/Pract16/welcome.jsp?uname=Manish

Welcome Manish second jsp page



localhost:8080/Pract16/second.jsp

Hello Manish

EXPERIMENT – XVII

AIM: **Demonstrate Action Cookie in JSP**

CODE:

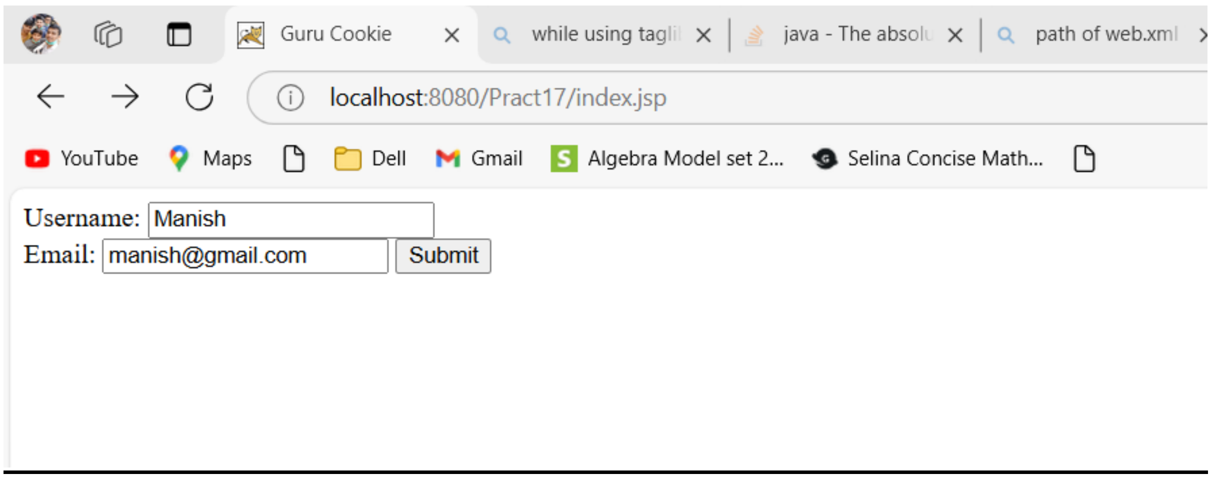
Index.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
   pageEncoding="ISO-8859-1"%>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Guru Cookie</title>
</head>
<body>
<form action="action_cookie_main.jsp" method="GET">
Username: <input type="text" name="username">
<br />
Email: <input type="text" name="email" />
<input type="submit" value="Submit" />
</form>
</body>
</html>
```

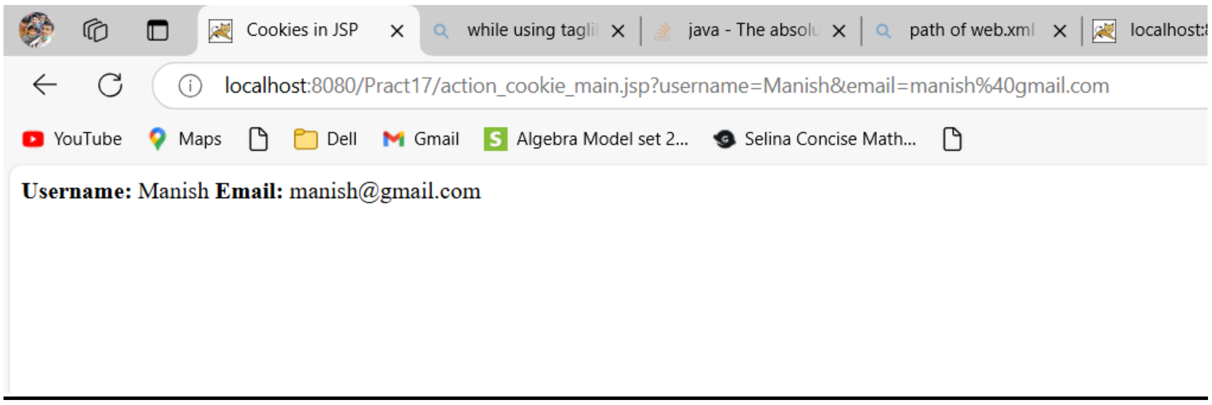
action_cookie_main

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
   pageEncoding="ISO-8859-1"%>

<%
  Cookie username = new Cookie("username",
 request.getParameter("username"));
  Cookie email = new Cookie("email",
  request.getParameter("email"));
  username.setMaxAge(60*60*10);
  email.setMaxAge(60*60*10);
  // Add both the cookies in the response header.
  response.addCookie( username );
  response.addCookie( email );
%>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Cookies in JSP</title>
</head>
<body>

<b>Username:</b>
<%= request.getParameter("username")%>
<b>Email:</b>
<%= request.getParameter("email")%>

</body>
</html>
```

<u>OUTPUT:</u>



Username: Manish
Email: manish@gmail.com Submit



```
Username: Manish Email: manish@gmail.com
```

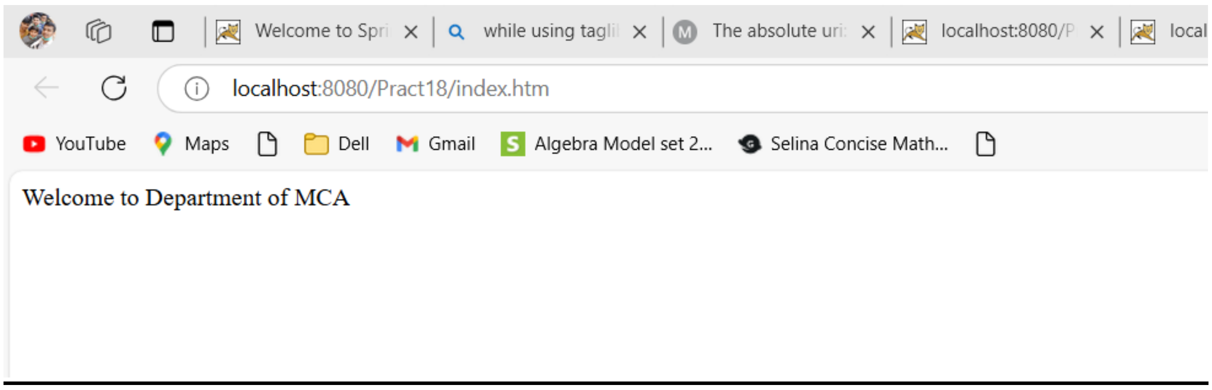
EXPERIMENT – XVIII

AIM: Demonstrate JSTL Core tag(c:out tag)

CODE:

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<html>
<head>
<title>Tag Example</title>
</head>
<body>
<c:out value="${'Welcome to Department of MCA '}"/>
</body>
</html>
```

OUTPUT:



```
Welcome to Department of MCA
```

EXPERIMENT – XIX

AIM: Demonstrate Dependency Injection Implementation

CODE:

Student.java

```
package com.jdbc.springjdbc.student;
public class Student {
    private String name;
    private int age;

    // Getters and setters
    public void setName(String name) {
        this.name = name;
    }
    public void setAge(int age) {
        this.age = age;
    }
    // Method to display student details
    public void show() {
        System.out.println("Student Name: " + name + ", Age: " + age);
    }
}
```

applicationContext,xml

```
package com.jdbc.springjdbc.student;

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

public class Test {

   public static void main(String[] args) {
        // Load the application context from XML
        ApplicationContext context = new ClassPathXmlApplicationContext("applicationContext.xml");

        // Retrieve the Student bean
        Student student = (Student) context.getBean("s1");

        // Use the bean
        student.show();
    }
}
```

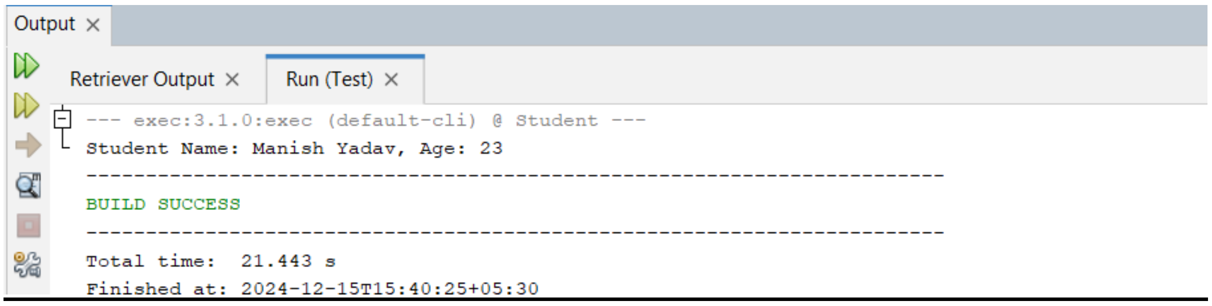
Test,java

```
package com.jdbc.springjdbc.student;

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

public class Test {

    public static void main(String[] args) {
        // Load the application context from XML
        ApplicationContext context = new ClassPathXmlApplicationContext("applicationContext.xml");

        // Retrieve the Student bean
        Student student = (Student) context.getBean("s1");

        // Use the bean
        student.show();
    }
}
```

Pom.xml

```
<dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring-context</artifactId>
        <version>5.3.29</version> <!-- Use a compatible version -->
    </dependency>
    <dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring-beans</artifactId>
        <version>5.3.29</version>
    </dependency>
    <dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring-core</artifactId>
        <version>5.3.29</version>
 </dependency>
```

OUTPUT:



```
--- exec:3.1.0:exec (default-cli) @ Student ---
Student Name: Manish Yadav, Age: 23
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  21.443 s
Finished at: 2024-12-15T15:40:25+05:30
```

EXPERIMENT – XX

AIM: **JDBC Data Access with Spring using MySQL / Oracle database**

CODE:

Database Table Creation:

```
CREATE TABLE Customer (
   ID   INT NOT NULL AUTO_INCREMENT,
   NAME VARCHAR(20) NOT NULL,
   AGE  INT NOT NULL,
   PRIMARY KEY (ID));
```

App.java

```
package com.jdbc.springjdbc.jdbc;


/**
 * Hello world!
 */
import java.util.List;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
import com.jdbc.springjdbc.jdbc.CustomerJDBCTemplate;


public class App {
   public static void main(String[] args) {
      ApplicationContext context = new ClassPathXmlApplicationContext("Beans.xml");


CustomerJDBCTemplate customerJDBCTemplate = (CustomerJDBCTemplate)
         context.getBean("customerJDBCTemplate");


      System.out.println("------Records Creation--------" );
customerJDBCTemplate.insert("Vipul", 40);
customerJDBCTemplate.insert("Jatin", 37);
customerJDBCTemplate.insert("Harpreet", 39);
//customerJDBCTemplate.delete(3);
      //customerJDBCTemplate.update(5,24);
   }
}
```

<u>Customer.java</u>

```
public class Customer {
   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;
   }
```

CustomerJDBCTemplate.java

```
import javax.sql.DataSource;
import org.springframework.jdbc.core.JdbcTemplate;
public class CustomerJDBCTemplate
{
  private DataSource dataSource;
  private JdbcTemplate jdbcTemplateObject;

  public void setDataSource(DataSource dataSource)
{
    this.dataSource = dataSource;
    this.jdbcTemplateObject = new JdbcTemplate(dataSource);
  }
  public void insert(String name, Integer age)
{
    String SQL = "insert into Customer (name,age) values('"+name+"','"+age+"')";

    jdbcTemplateObject.update( SQL);
    System.out.println("Created Record Name = " + name + " Age = " + age);
    return;
  }

  public void delete(int id)
{
    String SQL = "delete from Customer where id='"+id+"'";

    int n=jdbcTemplateObject.update( SQL);
    if(n>0)
    {
    System.out.println("Deleted Customer with ID = " + id);
    }
    return;
  }
   public void updateAge(Integer age, Integer id)
  {
        String SQL = "update Customer age='"+age+"' where id='"+id+"'";

        jdbcTemplateObject.update( SQL);
        System.out.println("Record Updated");
        return;
     }
}
```

Beans.xml

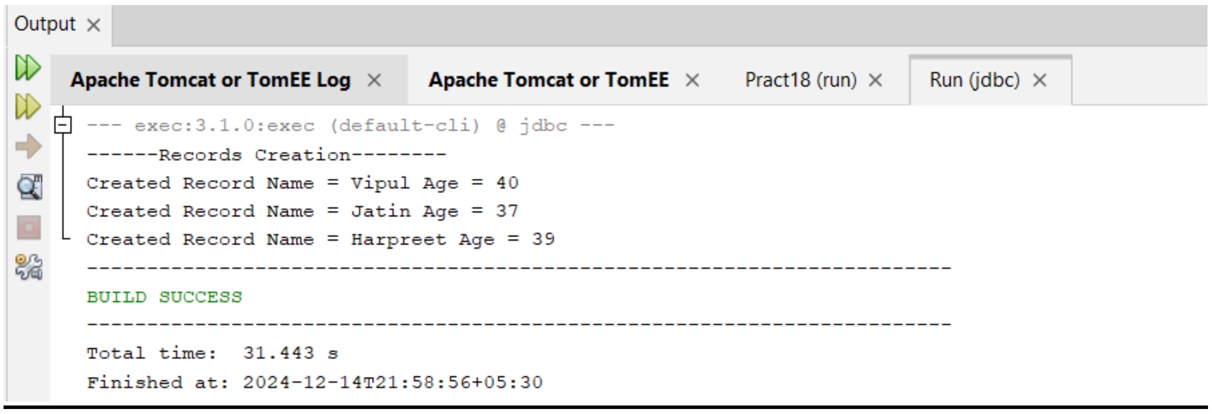
```
<?xml version = "1.0" encoding = "UTF-8"?>
<beans xmlns = "http://www.springframework.org/schema/beans"
   xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation = "http://www.springframework.org/schema/beans
   http://www.springframework.org/schema/beans/spring-beans-3.0.xsd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

<!-- Initialization for data source -->

<bean id = "customerJDBCTemplate" class = "com.jdbc.springjdbc.jdbc.CustomerJDBCTemplate">
<property name = "dataSource" ref = "dataSource" />
</bean>

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

<u>Pom.xml</u>

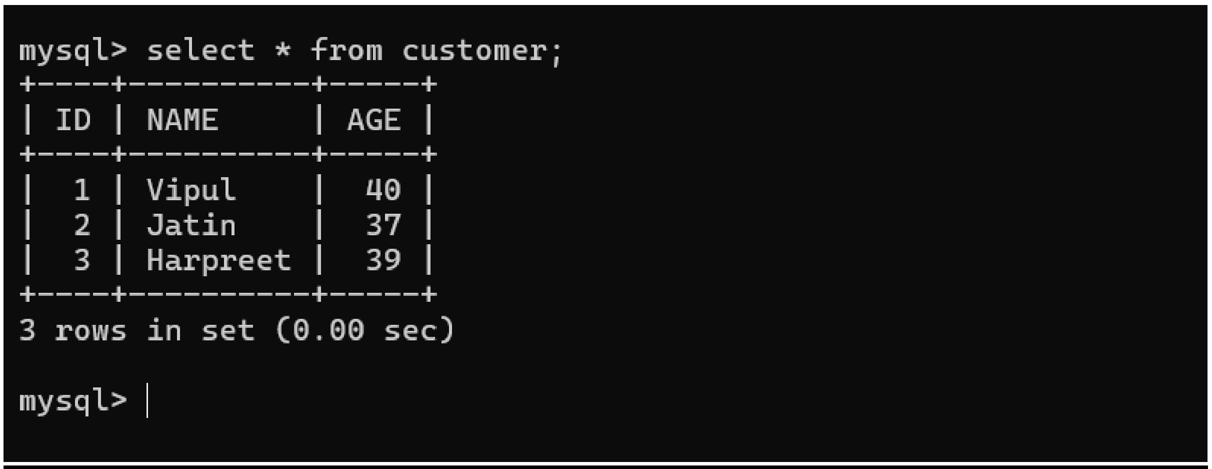
Under <dependencies> tag place the below code

```
<dependency>
            <groupId>org.springframework</groupId>
            <artifactId>spring-context</artifactId>
            <version>6.2.0</version>
        </dependency>

        <dependency>
            <groupId>org.springframework</groupId>
            <artifactId>spring-jdbc</artifactId>
            <version>6.2.0</version>
        </dependency>

        <dependency>
            <groupId>mysql</groupId>
            <artifactId>mysql-connector-java</artifactId>
            <version>8.0.17</version>
        </dependency>
```

OUTPUT:



```
--- exec:3.1.0:exec (default-cli) @ jdbc ---
------Records Creation--------
Created Record Name = Vipul Age = 40
Created Record Name = Jatin Age = 37
Created Record Name = Harpreet Age = 39
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  31.443 s
Finished at: 2024-12-14T21:58:56+05:30
```



```
mysql> select * from customer;
+----+----------+-----+
| ID | NAME     | AGE |
+----+----------+-----+
|  1 | Vipul    |  40 |
|  2 | Jatin    |  37 |
|  3 | Harpreet |  39 |
+----+----------+-----+
3 rows in set (0.00 sec)

mysql>
```