



ADVANCED JAVA (2160707)

LABORATORY MANUAL

B.E. Semester-VI

Computer Engineering
And
Information Technology

PREPARED BY:

CE/IT DEPARTMENT- 2019-2020

VADODARA INSTITUTE OF ENGINEERING, KOTAMBI-



Advance JAVA (2160707)

SR. NO	Experiments	Page No.
1.	Create an application using TCP to implement one way communication.	
2.	Create an application using TCP to implement two way communication.	
3.	Implement TCP Server for transferring files using Socket and ServerSocket.	
4.	User can create a new database and also create new table under that database. Once the database has been created then user can perform database operation by using Statement and PreparedStatement Interface.	
5.	Implement Student information system using JDBC Callable Statement.	
6.	Create a Servlet and retrieve Context and Config parameters.	
7.	Create Servlet file to insert, update and delete records of particular table of database.	
8.	Write a program to handle cookies.	
9.	Write a program to create filter which does pre and post processing.	
10.	Write a program to implement useBean action tag.	
11.	Write a program to create a Custom tag and use it in a JSP page.	
12.	Write a program using connectionless protocol in which the client sends a string to server and server converts the string into uppercase and sends it back to client.	
13.	Write a JSF application to create a login form.	
14.	Write a JSF application to insert, update and delete record from database.	

Software & Hardware Requirements

Sr.No	Software Requirement	Hardware Requirement
1.	Netbeans 6.9 or later	Processor : Pentium IV
2.	JDK 1.6 or later	RAM : 256 MB
3.	SQL plus (Oracle) 11g	Hard Disk : 40 GB
4.	MySQL 4.0 or Higher Version	Monitor : 15 “



Advance JAVA (2160707)

Practical-1

AIM: Create an application using TCP to implement one way communication.

Source Code:

SenderServer.java

```
package mynetworkapp;

import java.io.*;

import java.net.*;
import java.util.Scanner;

public class SenderServer {

    public static void main(String[] args) {
        // TODO code application logic here
        try{
            ServerSocket ss=new ServerSocket(4020);
            System.out.println("wating for client to connect");
            Socket s=ss.accept();

            System.out.println("connection established");

            PrintStream ps=new PrintStream(s.getOutputStream());
            BufferedReader kb=new BufferedReader(new InputStreamReader(System.in));

            System.out.println("Enter data :");
            String str=kb.readLine();

            while(!str.equals("exit"))
            {
                ps.println(str);
                System.out.println("Enter data :");
                str=kb.readLine();
            }
            ps.close();
            s.close();
            ss.close();

        }catch(IOException e)
        {
        }
```



Advance JAVA (2160707)

```
        System.out.println(e);
    }
}
}
```

ReceiverClient.java

```
package mynetworkapp;

import java.net.InetAddress;
import java.net.Socket;
import java.util.*;
import java.io.*;

public class ReceiverClient {

    public static void main(String[] args) {
        // TODO code application logic here
        try{

            // InetAddress ip=InetAddress.getLocalHost();

            Socket s=new Socket("localhost",4020);

            BufferedReader br=new BufferedReader(new InputStreamReader(s.getInputStream()));

            String str;

            str=br.readLine();

            while(str!=null)
            {
                System.out.println(str);
                str=br.readLine();
            }
            br.close();
            s.close();
        }catch(Exception e)
        {
            System.out.println(e);
        }

    }
}
```



Advance JAVA (2160707)

OUTPUT

Server Side

```
run:
connection established
Enter string
hello
BUILD SUCCESSFUL (total time: 15 seconds)
```

Client Side

```
run:
From serverhello
From serverBye
BUILD SUCCESSFUL (total time: 11 seconds)
```

Logical Questions:

1. How to establish connection between client and server?
2. How to send message from server to client only?



Advance JAVA (2160707)

Practical 2

Aim: Create an application using TCP to implement two way communication.

Source Code:

Server.java

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package mynetworkapp;

import java.io.*;

import java.net.*;

/**
 *
 * @author faraahdabhoiwala
 */
public class TwoWayServer1 {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        try{
            ServerSocket ss=new ServerSocket(4021);
            System.out.println("Waiting for client to connect");
            Socket s=ss.accept();

            System.out.println("connection established");

            BufferedReader br=new BufferedReader(new InputStreamReader(s.getInputStream()));
            BufferedReader kb=new BufferedReader(new InputStreamReader(System.in));
            PrintStream ps=new PrintStream(s.getOutputStream());

            String str,str1;
            str=br.readLine();//read frim client
```



Advance JAVA (2160707)

```
while(str!=null)
{
    System.out.println(str);//reads frm client n writes on server
    System.out.print("Write something : ");
    str1=kb.readLine();
    ps.println(str1);

    str=br.readLine();//reads frm client
}

ps.close();
br.close();
kb.close();
s.close();
ss.close();

} catch(IOException e)

{
    System.out.println(e);
}
}
```

Client.java

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package mynetworkapp;

import java.io.*;
import java.net.*;

/**
 * @author faraahdabhoiwala
 */
public class TwoWayClient1 {

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



Advance JAVA (2160707)

```
public static void main(String[] args) {
    // TODO code application logic here
    try{
        // InetAddress ip= InetAddress.getLocalHost();

        Socket s=new Socket("192.168.1.46",4021);
        //"192.168.1.47"

        BufferedReader br=new BufferedReader(new InputStreamReader(s.getInputStream()));
        BufferedReader kb=new BufferedReader(new InputStreamReader(System.in));
        PrintStream ps=new PrintStream(s.getOutputStream());

        System.out.print("write something : ");
        String str=kb.readLine();

        String str1;

        while(!(str.equals("exit")))
        {
            ps.println(str);

            str1=br.readLine();
            System.out.println(str1);
            System.out.print("write something : ");
            str=kb.readLine();

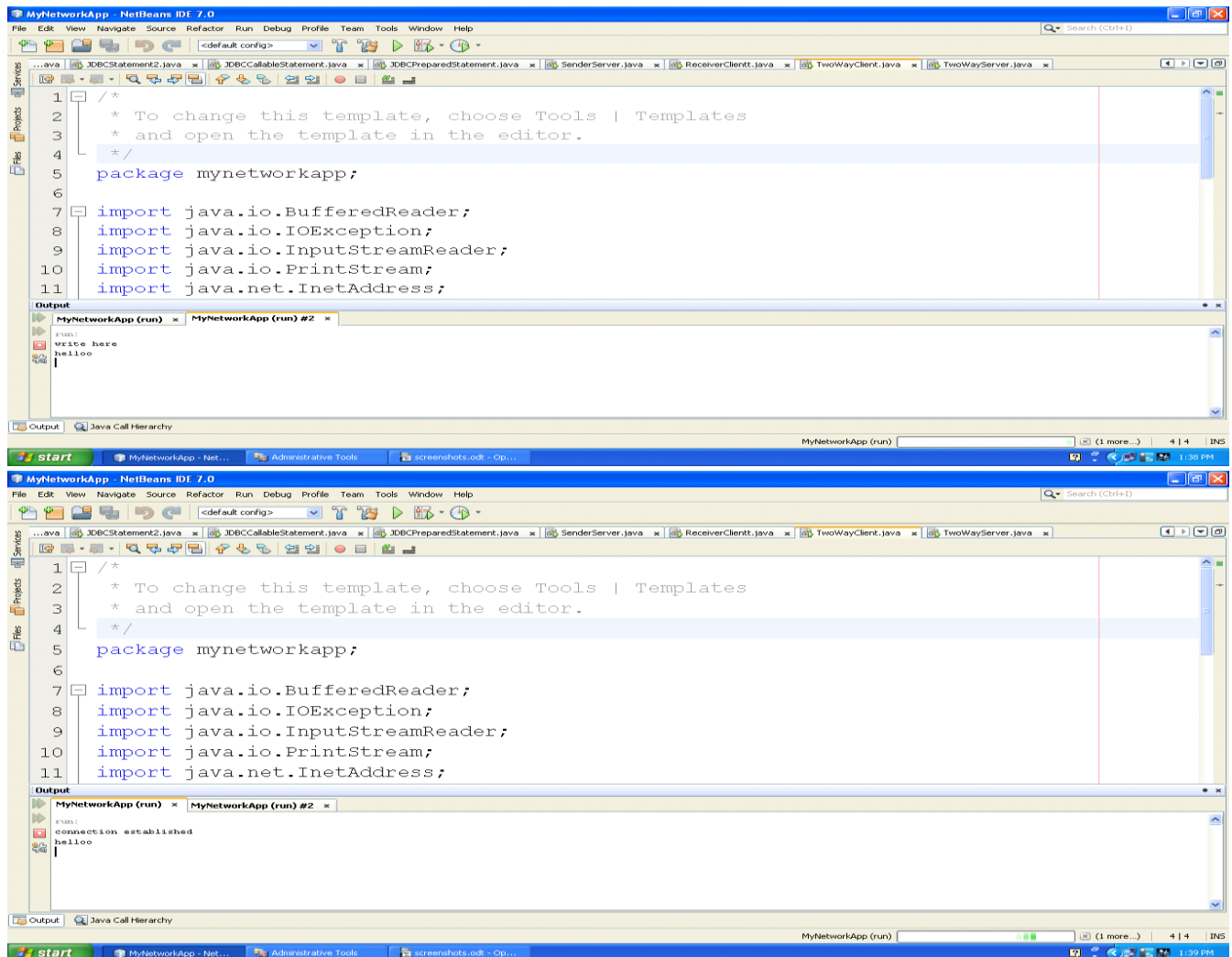
        }

        ps.close();
        br.close();
        kb.close();
        s.close();
        System.out.println("client program ended");

    }catch(IOException e)
    {
        System.out.println(e);
    }
}
```




Advance JAVA (2160707)



Logical Questions:

1. Write a program in which server sends a string and client reply with reverse of that string ?
2. Write a client server program using TCP where client sends a string and server checks whether that string is palindrome or not and responds with appropriate message.

Practical 3

Aim: Implement TCP Server for transferring files using Socket and ServerSocket.

Source Code:

FileServer.java

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package mynetworkapp;

import java.io.*;

import java.net.*;

/**
 *
 * @author faraahdabhoiwala
 */
public class FileServer {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        try{
            ServerSocket ss=new ServerSocket(4022);
            System.out.println("Wating for client to connect");
            Socket s=ss.accept();

            System.out.println("connection established");

            BufferedReader br=new BufferedReader(new InputStreamReader(s.getInputStream()));

            PrintStream ps=new PrintStream(s.getOutputStream());//client

            String fname;
            fname=br.readLine();

            File f=new File(fname);
            if(f.exists()==true)
```



Advance JAVA (2160707)

```
{
//FileReader fr=new FileReader(fname);
BufferedReader br1=new BufferedReader(new FileReader(fname));
String str;

str=br1.readLine();
while(str!=null)
{
    ps.println(str);
    str=br1.readLine();
}

br1.close();
}
else
{

    ps.println("file doesnt exist");
}
ps.close();
br.close();

s.close();
ss.close();

}catch(IOException e)

{
    System.out.println(e);
}
}
```

FileClient.java

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package mynetworkapp;

import java.io.*;
import java.net.InetAddress;
import java.net.Socket;

/**
```



Advance JAVA (2160707)

```
*
* @author faraahdabhoiwala
*/
public class FileClient {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        try{
            // InetAddress ip= InetAddress.getLocalHost();

            Socket s=new Socket("localhost",4022);
            //"192.168.1.47"

            BufferedReader br=new BufferedReader(new InputStreamReader(s.getInputStream()));
            BufferedReader kb=new BufferedReader(new InputStreamReader(System.in));
            PrintStream ps=new PrintStream(s.getOutputStream());

            System.out.print("Enter file name : ");
            String str=kb.readLine();
            ps.println(str);

            String st=br.readLine();
            while(st!=null)
            {
                System.out.println(st);
                st=br.readLine();
            }

            ps.close();
            br.close();
            kb.close();
            s.close();
            System.out.println("client program ended");

        }catch(IOException e)
        {
            System.out.println(e);
        }
    }
}
```



Advance JAVA (2160707)

OUTPUT:

```
1: /*
2: * To change this template, choose Tools | Templates
3: * and open the template in the editor.
4: */
5: package filetransferer;
6: import java.io.*;
7: import java.net.*;
8: /**
9: *
10: * @author krutikoshiya
11: */
```

Output: MyNetworkApp (run) #2

run: 1: test: 2: enter file name 3: V:\k.txt 4: hello from server 5: BUILD SUCCESSFUL (total time: 1 minute 27 seconds)

```
1: /*
2: * To change this template, choose Tools | Templates
3: * and open the template in the editor.
4: */
5: package filetransferer;
6: import java.io.*;
7: import java.net.*;
8: /**
9: *
10: * @author krutikoshiya
11: */
```

Output: MyNetworkApp (run) #2

run: 1: connection established 2: BUILD SUCCESSFUL (total time: 1 minute 31 seconds)

Logical Questions:

- 1. Write a TCP program to access the contents of file at client side.*
- 2. Which classes are used for file transfer in socket programming?*



Advance JAVA (2160707)

Practical 4

Aim: User can create a new database and also create new table under that database. Once the database has been created then user can perform database operation by using Statement and PreparedStatement Interface.

Source Code:

```
package myfirstapp;
import java.sql.*;

public class JDBCStatement {

    public static void main(String[] args) {
        // TODO code application logic
        try
        {
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

            Connection
con=DriverManager.getConnection("jdbc:odbc:TempDataSource","system","system");
            Statement st=con.createStatement();

            ResultSet rs=st.executeQuery("select * from student");

            while(rs.next())
            {

                System.out.println(rs.getString("id") + " "+rs.getString("s_name")+ " "+
rs.getString("address"));

            }

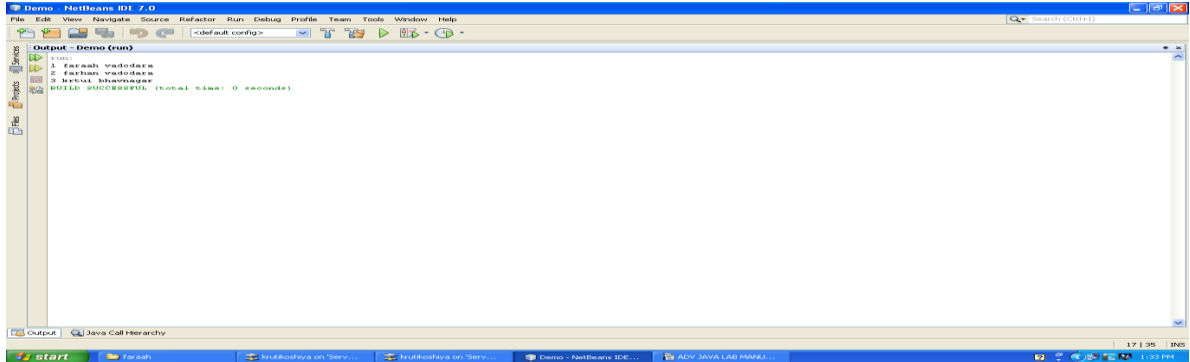
        }catch(SQLException e)
        {

            System.out.print(e);
        }
        catch(ClassNotFoundException e)
        {
            System.out.print(e);
        }
    }
}
```



Advance JAVA (2160707)

OUTPUT:



```
package myfirstapp;

import java.sql.*;

public class JDBCStatement2 {

    public static void main(String[] args) {
        // TODO code application logic
        int id=6;
        String name ="faraah";
        String address="vadodara";

        try
        {
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
            Connection
con=DriverManager.getConnection("jdbc:odbc:TempDataSource","system","system");
            Statement st=con.createStatement();
            String query="insert into f_student values("+id+", '"+name+"', '"+address+"')";
            int count=st.executeUpdate(query);
            System.out.println("No of rows updated="+count);

        }catch(Exception e)
        {
```



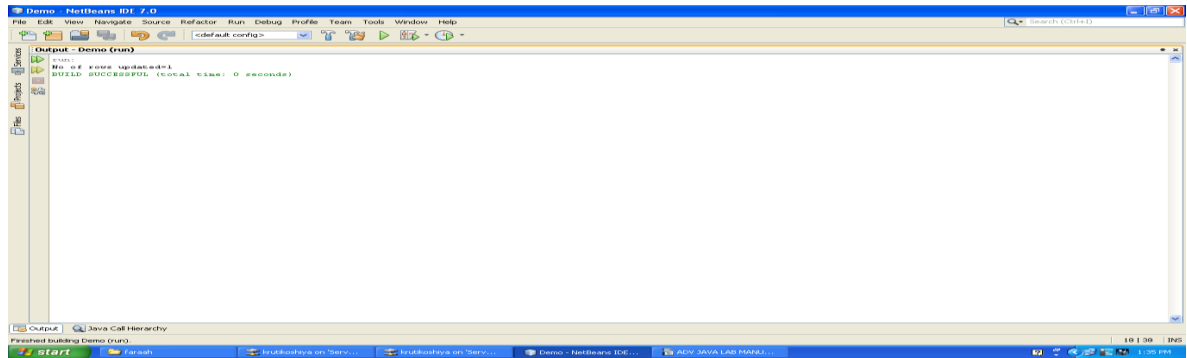
Advance JAVA (2160707)

```
        System.out.print(e);
    }

}

}
```

OUTPUT:



```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package myfirstapp;

import java.sql.*;

/**
 *
 * @author faraahdabhoiwala
 */
public class JDBCPreparedStatement {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        try
        {
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
            Connection
con=DriverManager.getConnection("jdbc:odbc:TempDataSource","system","system");
```




Advance JAVA (2160707)

```
PreparedStatement ps=con.prepareStatement("select * from f_student where name like ? or name like ?");
```

```
ps.setString(1, "r%");  
ps.setString(2, "j%");
```

```
ResultSet rs=ps.executeQuery();  
while(rs.next())  
{
```

```
    System.out.println(rs.getString("id")+ " "+rs.getString("name")+ " "+rs.getString("address"));  
}
```

```
    }catch(Exception e)  
    {  
        System.out.print(e);  
    }  
}
```

Using PreparedStatement Interface:-

```
/*  
 * To change this template, choose Tools | Templates  
 * and open the template in the editor.  
 */  
package myfirstapp;  
  
import java.sql.*;  
  
/**  
 *  
 * @author faraahdabhoiwala  
 */  
public class JDBCPreparedStatement {  
  
    /**  
     * @param args the command line arguments  
     */  
    public static void main(String[] args) {  
        // TODO code application logic here  
        try
```



Advance JAVA (2160707)

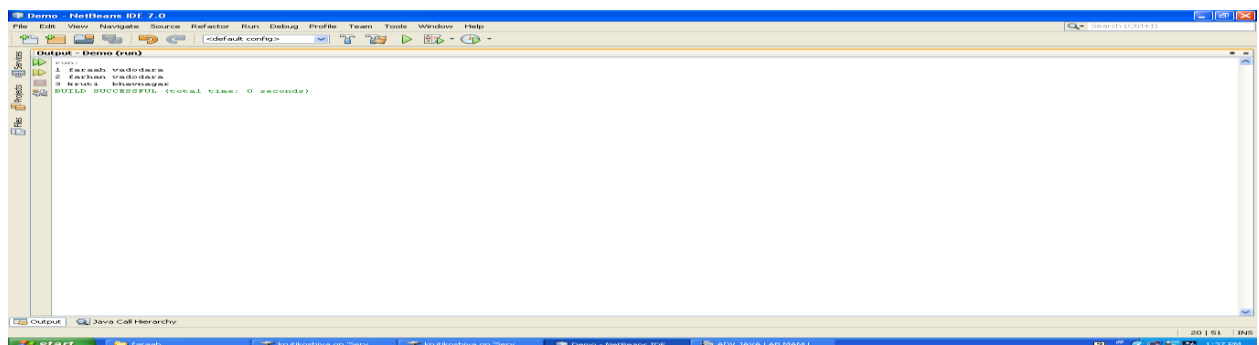
```
{
    Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
    Connection
con=DriverManager.getConnection("jdbc:odbc:TempDataSource","system","system");

    PreparedStatement ps=con.prepareStatement("select * from f_student name like ? or name like
?");
    ps.setString(1, "k%");
    ps.setString(2, "f%");

    ResultSet rs=ps.executeQuery();
    while(rs.next())
    {
        System.out.println(rs.getString("id")+ " "+rs.getString("name")+ " "+rs.getString("address"));
    }

    }catch(Exception e)
    {
        System.out.print(e);
    }
}
```

OUTPUT:



Logical Questions:

1. What is the difference between statement and prepared statement?
2. How prepared statement is better than statement interface?
3. Insert the data in the database by using prepared statement interface.



Advance JAVA (2160707)

Practical 5

Aim: Implement Student information system using JDBC Callable Statement.

Source Code:

```
package myfirstapp;

import java.sql.*;
import java.util.Scanner;

/**
 *
 * @author faraahdabhoiwala
 */
public class JDBCCallableStatement {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here

        try
        {
            //Scanner sc=new Scanner(System.in);
            //System.out.print("enter name");

            String name="vrushil11";
            int enr=13;
            int sem=5;
            String add="kotambi";
            int per=99;

            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

            Connection
con=DriverManager.getConnection("jdbc:odbc:TempDataSource","system","system");

            CallableStatement cs=con.prepareCall("call add_student_details (?, ?, ?, ?, ?)");

            cs.setInt(1, enr);

            cs.setString(2, name);
            cs.setInt(3, sem);
```



Advance JAVA (2160707)

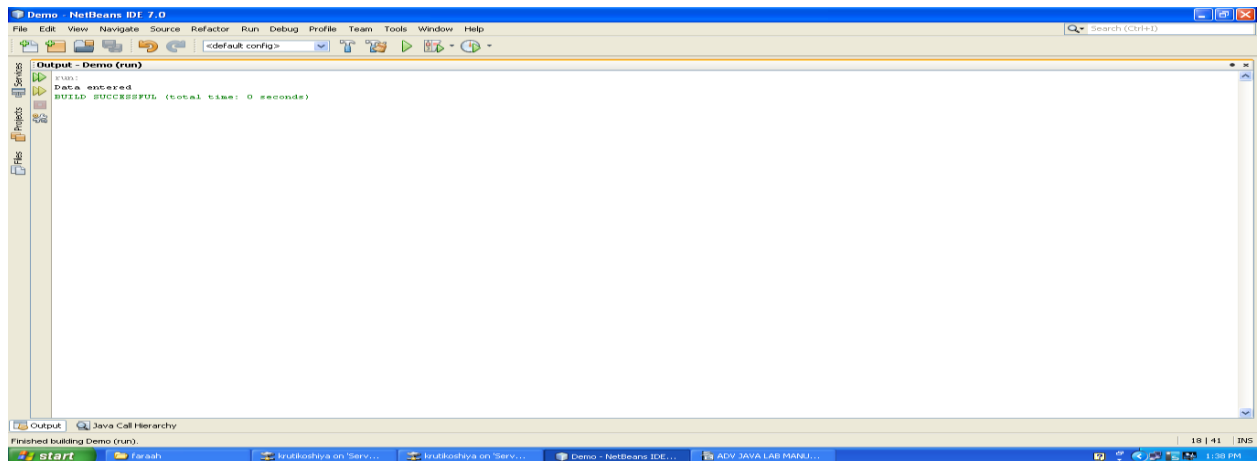
```
cs.setString(4, add);
cs.setInt(5, per);

cs.execute();

System.out.println("Data entered");

} catch(SQLException e)
{
    System.out.print(e);
}
catch(ClassNotFoundException e)
{
    System.out.print(e);
}
}
```

OUTPUT:



Using OUT parameters in CallableStatement Interface:-

package myfirstapp;

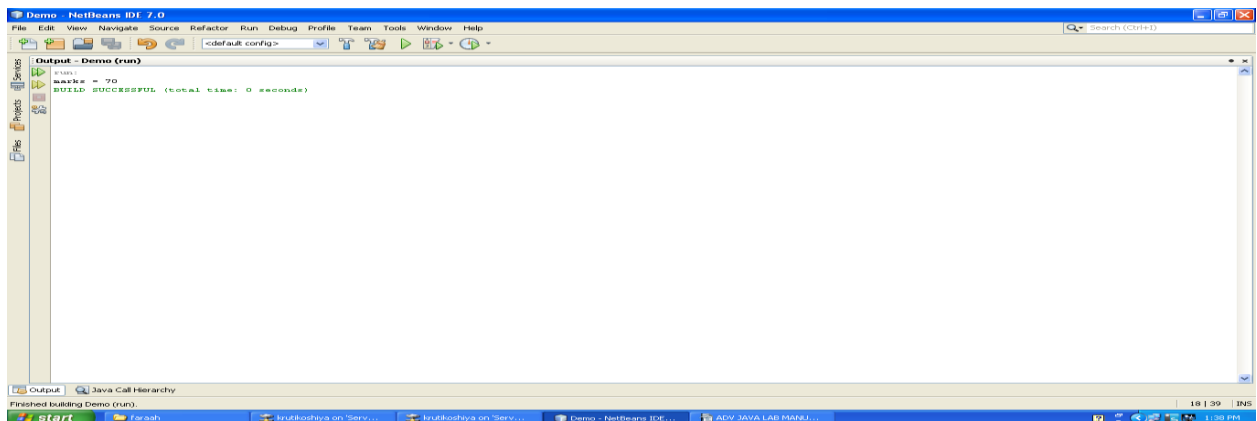
```
import java.sql.*;
import java.util.Scanner
/**
 *
 * @author faraahdabhoiwala
 */
public class JDBCCallableOUT
/**
 * @param args the command line arguments
 */
```



Advance JAVA (2160707)

```
public static void main(String[] args) {  
    // TODO code application logic  
  
    Scanner sc=new Scanner(System.in);  
    System.out.println("Enter id :");  
    int id=sc.nextInt();  
    try {  
        Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  
  
        Connection con=DriverManager.getConnection("jdbc:odbc:TempDataSource","system","system");  
  
        CallableStatement cs=con.prepareCall("{ call getStudentMarks(?,?) }");  
        cs.setInt(1,id);  
        cs.registerOutParameter(2, Types.INTEGER);  
        cs.execute();  
        int percentage=cs.getInt(2);  
        System.out.println("marks =" +percentage);  
  
    } catch (Exception e) {  
        System.out.println(e);  
    }  
}
```

OUTPUT:



Logical Questions:

1. What is Stored Procedure?
2. How to execute more than one query by using callable statement interface?
3. Update the record in the database by using callable statement interface.



Advance JAVA (2160707)

Practical 6

Aim: Create a Servlet and retrieve Context and Config parameters.

Source Code:

index.jsp

```
<%--
    Document : index
    Created on : Feb 16, 2016, 10:37:58 AM
    Author : faraahdabhoiwala
--%>

<% @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>
        <a href="RetriveParam"> Go to MyServlet</a>
    </body>
</html>
```

WEB-INF

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="3.0" xmlns="http://java.sun.com/xml/ns/javaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-
app_3_0.xsd">
    <context-param>
        <param-name>admin</param-name>
        <param-value>dhaval</param-value>
    </context-param>
    <context-param>
        <param-name>user</param-name>
        <param-value>saifali</param-value>
    </context-param>
    <servlet>
        <servlet-name>RetriveParam</servlet-name>
        <servlet-class>RetriveParam</servlet-class>
        <init-param>
            <param-name>country</param-name>
            <param-value>india</param-value>
```



Advance JAVA (2160707)

```
</init-param>
<init-param>
  <param-name>college</param-name>
  <param-value>vier</param-value>
</init-param>
<init-param>
  <param-name>s_name</param-name>
  <param-value>nirlay</param-value>
</init-param>
</servlet>
<servlet-mapping>
  <servlet-name>RetriveParam</servlet-name>
  <url-pattern>/RetriveParam</url-pattern>
</servlet-mapping>
<session-config>
  <session-timeout>
    30
  </session-timeout>
</session-config>
</web-app>
```

RetriveParam.java

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletConfig;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.*;
import java.util.*;

/**
 *
 * @author faraahdabhoiwala
 */
public class RetriveParam extends HttpServlet {

    /**
     * Processes requests for both HTTP GET and POST methods.
     * @param request servlet request
     * @param response servlet response
     */
}
```



Advance JAVA (2160707)

```
* @throws ServletException if a servlet-specific error occurs
* @throws IOException if an I/O error occurs
*/
protected void processRequest(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");
    PrintWriter out = response.getWriter();
    ServletConfig sc=this.getServletConfig();

    Enumeration e=sc.getInitParameterNames();
    while(e.hasMoreElements())
    {
        String str=(String)e.nextElement();//name of param
        out.print("<br><b>" +sc.getInitParameter(str));//value
    }

    ServletContext sc1=this.getServletContext();

    Enumeration e1=sc1.getInitParameterNames();
    while(e1.hasMoreElements())
    {
        String str1=(String)e1.nextElement();//name of param
        out.print("<br>" +str1);
        out.print("<br><b>" +sc1.getInitParameter(str1));//value
    }
}

// <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on the left to
edit the code.">
/**
 * Handles the HTTP <code>GET</code> method.
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    processRequest(request, response);
}

/**
 * Handles the HTTP <code>POST</code> method.
```




Advance JAVA (2160707)

```
* @param request servlet request
* @param response servlet response
* @throws ServletException if a servlet-specific error occurs
* @throws IOException if an I/O error occurs
*/
@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    processRequest(request, response);
}

/**
 * Returns a short description of the servlet.
 * @return a String containing servlet description
 */
@Override
public String getServletInfo() {
    return "Short description";
} // </editor-fold>
}
```

OUTPUT:

dhaval

saifali

india

vier

nirlay

Logical Questions:

1. Write a program for getting all the initialization parameter from the web.xml file and printing this information in the servlet.

2. Which methods are used for ServletContext and ServletConfig?



Advance JAVA (2160707)

Practical 7

Aim: Create Servlet file to insert, update and delete records of particular table of database.

Source Code:

Student.html

```
<!--
To change this template, choose Tools | Templates
and open the template in the editor.
-->
<!DOCTYPE html>
<html>
  <head>
    <title></title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  </head>
  <body>

    <form action="JDBCServlet">
      Name<input type="text" name="name"></input>
      <br>
      Enrollment NO<input type="text" name="enr"></input>
      <br>
      Semester<input type="text" name="sem"></input>
      <br>
      <input type="submit" value="insert" name="submit"></input>

      <input type="submit" value="update" name="submit"></input>
      <input type="submit" value="delete" name="submit"></input>
    </form>
  </body>
</html>
```

JDBCServlet.java

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.sql.*;
```



Advance JAVA (2160707)

```
/**
 *
 * @author faraahdabhoiwala
 */
public class JDBCServlet extends HttpServlet {
    int enr,sem;
    String name;

    /**
     * Processes requests for both HTTP GET and POST methods.
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();

        enr=Integer.parseInt(request.getParameter("enr"));
        name=request.getParameter("name");
        sem=Integer.parseInt(request.getParameter("sem"));

        String str=request.getParameter("submit");
        if(str.equals("insert"))
        {

            String s=insertData();

            out.println("<br> "+s);
        }
        else if(str.equals("update"))
        {

            updateData();

            out.println("<br> data updated successfully");
        }
        else if(str.equals("delete"))
        {
            deleteData();
            out.println("<br> data deleted successfully");
        }
    }
}
```



Advance JAVA (2160707)

```
public String insertData()
{ try
{
    Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
    Connection
con=DriverManager.getConnection("jdbc:odbc:TempDataSource","system","system");
    Statement st=con.createStatement();
    String query="insert into f_student values('+enr+', '"+name+', '+sem+')";

    int count=st.executeUpdate(query);
    return "data inserted succesfully";
} catch(Exception e)
{
    return e.toString();
}
}
public void updateData()
{ try
{
    Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
    Connection
con=DriverManager.getConnection("jdbc:odbc:TempDataSource","system","system");
    Statement st=con.createStatement();
    String query="update f_student set name='"+name+"' where enr='"+enr+"'";
    int count=st.executeUpdate(query);

} catch(Exception e)
{
    System.out.print(e);
}
}
public void deleteData()
{ try
{
    Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
    Connection
con=DriverManager.getConnection("jdbc:odbc:TempDataSource","system","system");
    Statement st=con.createStatement();
    String query="delete from f_student where enr='"+enr+"'";
    int count=st.executeUpdate(query);

} catch(Exception e)
{
    System.out.print(e);
}
}
```



Advance JAVA (2160707)

// <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on the left to edit the code.">

```
/**
 * Handles the HTTP <code>GET</code> method.
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    processRequest(request, response);
}

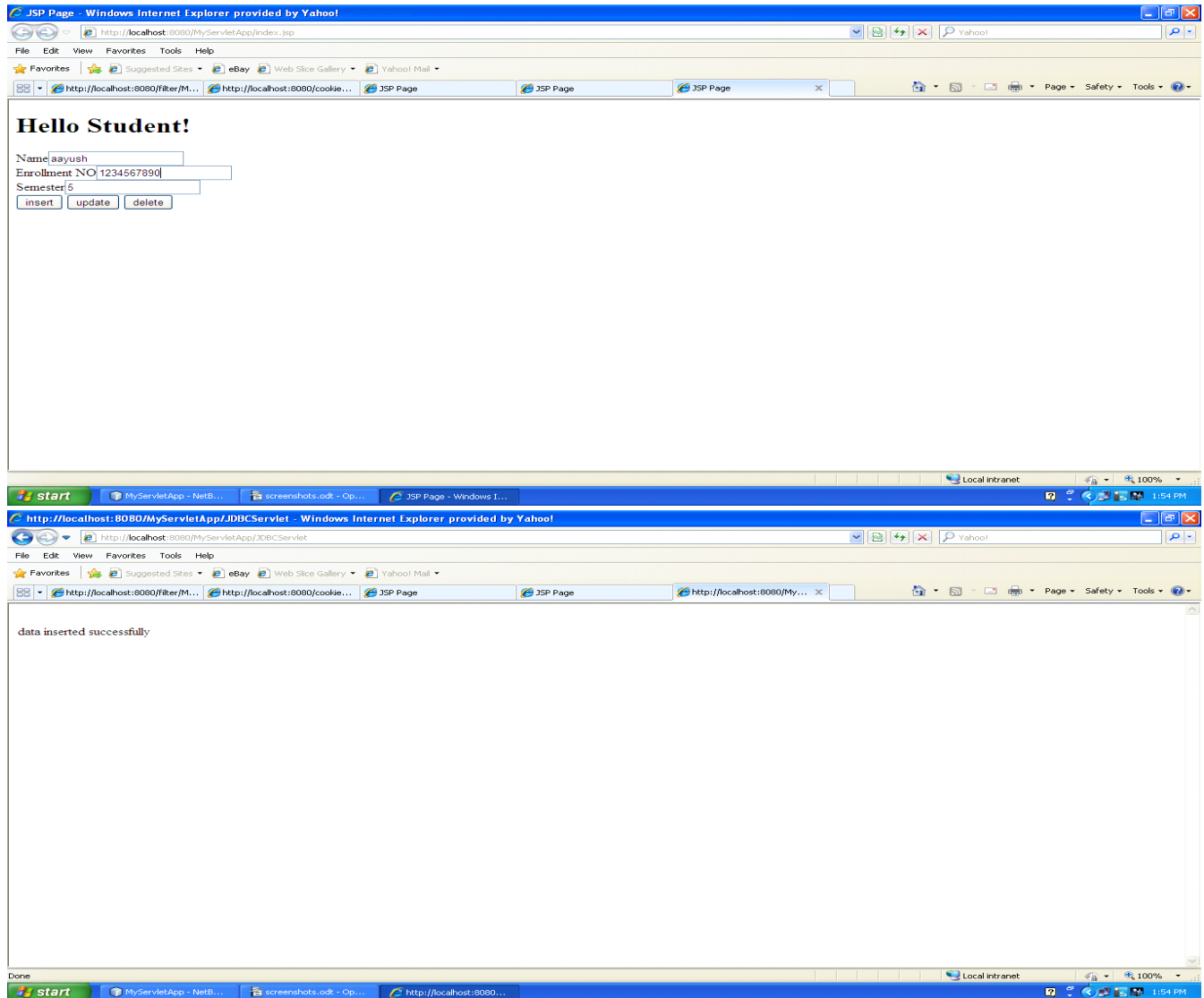
/**
 * Handles the HTTP <code>POST</code> method.
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    processRequest(request, response);
}

/**
 * Returns a short description of the servlet.
 * @return a String containing servlet description
 */
@Override
public String getServletInfo() {
    return "Short description";
} // </editor-fold>
}
```



Advance JAVA (2160707)

OUTPUT:



Logical Questions:

1. How we can insert the record in the database by using servlet?
2. How we can update the record in the database by using servlet?



Advance JAVA (2160707)

Practical 8

Aim: Write a program to handle cookies.

Source Code:

SessionCookies.jsp

```
<%--
    Document : SessionCookies
    Created on : Feb 19, 2016, 11:57:27 AM
    Author : faraahdabhoiwala
--%>

<% @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>
        <a href="SessionCookies">Retrive My Cookies</a>

    </head>
    <body>
        <h1>Hello World!</h1>
    </body>
</html>
```

SessionCookies.java

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.Cookie;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

/**
 *
 * @author faraahdabhoiwala
```



Advance JAVA (2160707)

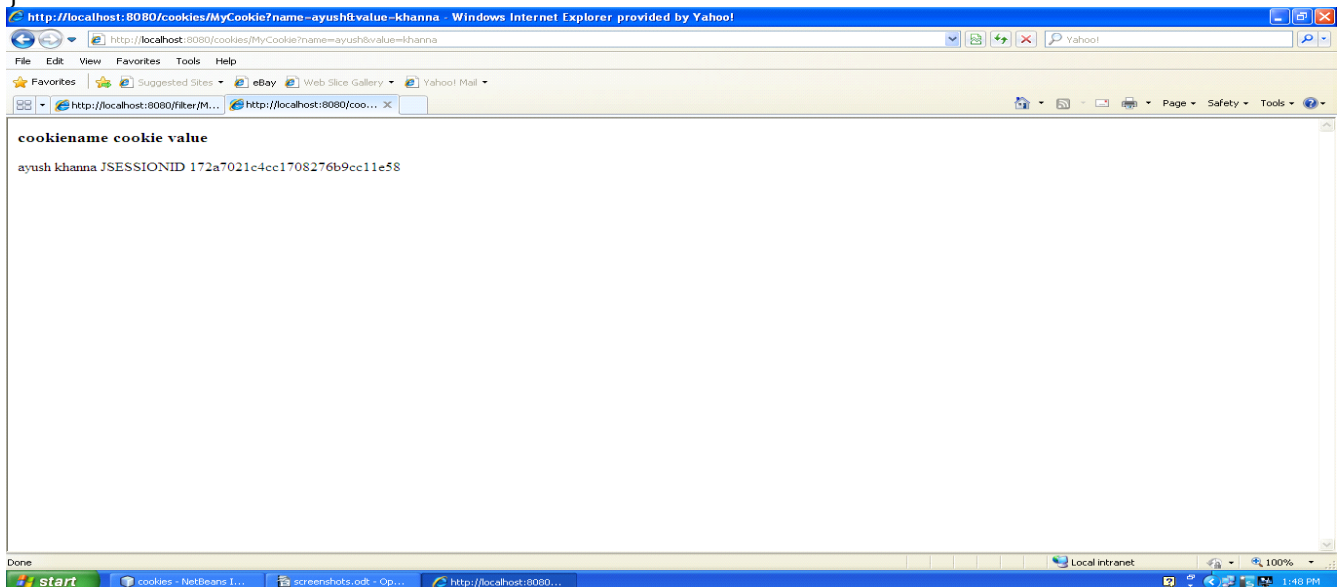
```
*/  
public class SessionCookies extends HttpServlet {  
  
    /**  
    * Processes requests for both HTTP GET and POST methods.  
    * @param request servlet request  
    * @param response servlet response  
    * @throws ServletException if a servlet-specific error occurs  
    * @throws IOException if an I/O error occurs  
    */  
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)  
        throws ServletException, IOException {  
        response.setContentType("text/html;charset=UTF-8");  
        PrintWriter out = response.getWriter();  
  
        Cookie c1=new Cookie("username","faraah");  
        Cookie c2=new Cookie("password","f1234");  
  
        response.addCookie(c1);  
        response.addCookie(c2);  
        Cookie[] c=request.getCookies();  
        for(int i=0;i<c.length;i++)  
        {  
            out.println("<br>name of cookie is"+c[i].getName()+"value="+c[i].getValue());  
        }  
    }  
  
    // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on the left to  
    edit the code.">  
    /**  
    * Handles the HTTP GET method.  
    * @param request servlet request  
    * @param response servlet response  
    * @throws ServletException if a servlet-specific error occurs  
    * @throws IOException if an I/O error occurs  
    */  
    @Override  
    protected void doGet(HttpServletRequest request, HttpServletResponse response)  
        throws ServletException, IOException {  
        processRequest(request, response);  
    }  
  
    /**  
    * Handles the HTTP POST method.
```




Advance JAVA (2160707)

```
* @param request servlet request
* @param response servlet response
* @throws ServletException if a servlet-specific error occurs
* @throws IOException if an I/O error occurs
*/
@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    processRequest(request, response);
}

/**
 * Returns a short description of the servlet.
 * @return a String containing servlet description
 */
@Override
public String getServletInfo() {
    return "Short description";
} // </editor-fold>
```



Logical Questions:

1. To add the cookie in the browser, which method is used?
2. How to retrieve the cookie value on browser?



Advance JAVA (2160707)

Practical 9

Aim: Write a program to create filter which does pre and post processing.

Source Code:

newhtml.html

```
<!--  
To change this template, choose Tools | Templates  
and open the template in the editor.  
-->  
<!DOCTYPE html>  
<html>  
  <head>  
    <title></title>  
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">  
  </head>  
  <body>  
    <div>TODO write content</div>  
    <a href="testServlet">go to my filter</a>  
  </body>  
</html>
```

MyFilter.java

```
/*  
 * To change this template, choose Tools | Templates  
 * and open the template in the editor.  
 */  
  
import java.io.IOException;  
import java.io.InputStream;  
import java.io.PrintWriter;  
import java.io.StringWriter;  
import javax.servlet.Filter;  
import javax.servlet.FilterChain;  
import javax.servlet.FilterConfig;  
import javax.servlet.ServletException;  
import javax.servlet.ServletRequest;  
import javax.servlet.ServletResponse;  
  
/**  
 *  
 * @author faraahdabhoiwala  
 */  
public class MyFilter implements Filter {  
  
    private static final boolean debug = true;  
    // The filter configuration object we are associated with. If
```



Advance JAVA (2160707)

```
// this value is null, this filter instance is not currently
// configured.
private FilterConfig filterConfig = null;

public MyFilter() {
}

private void doBeforeProcessing(ServletRequest request, ServletResponse response)
    throws IOException, ServletException {
    if (debug) {
        log("MyFilter:DoBeforeProcessing");
    }

    // Write code here to process the request and/or response before
    // the rest of the filter chain is invoked.

    // For example, a logging filter might log items on the request object,
    // such as the parameters.
    /*
    for (Enumeration en = request.getParameterNames(); en.hasMoreElements(); ) {
        String name = (String)en.nextElement();
        String values[] = request.getParameterValues(name);
        int n = values.length;
        StringBuffer buf = new StringBuffer();
        buf.append(name);
        buf.append("=");
        for(int i=0; i < n; i++) {
            buf.append(values[i]);
            if (i < n-1)
                buf.append(",");
        }
        log(buf.toString());
    }
    */
}

private void doAfterProcessing(ServletRequest request, ServletResponse response)
    throws IOException, ServletException {
    if (debug) {
        log("MyFilter:DoAfterProcessing");
    }

    // Write code here to process the request and/or response after
    // the rest of the filter chain is invoked.

    // For example, a logging filter might log the attributes on the
    // request object after the request has been processed.
```



Advance JAVA (2160707)

```
/*
for (Enumeration en = request.getAttributeNames(); en.hasMoreElements(); ) {
String name = (String)en.nextElement();
Object value = request.getAttribute(name);
log("attribute: " + name + "=" + value.toString());

}
*/

// For example, a filter might append something to the response.
/*
PrintWriter respOut = new PrintWriter(response.getWriter());
respOut.println("<P><B>This has been appended by an intrusive filter.</B>");
*/
}

/**
 *
 * @param request The servlet request we are processing
 * @param response The servlet response we are creating
 * @param chain The filter chain we are processing
 *
 * @exception IOException if an input/output error occurs
 * @exception ServletException if a servlet error occurs
 */
public void doFilter(ServletRequest request, ServletResponse response,
    FilterChain chain)
    throws IOException, ServletException {

    if (debug) {
        log("MyFilter:doFilter()");
    }

    doBeforeProcessing(request, response);
    PrintWriter out=response.getWriter();
    out.print("<br>hi from filter");
    Throwable problem = null;
    try {
        chain.doFilter(request, response);
    } catch (Throwable t) {
        // If an exception is thrown somewhere down the filter chain,
        // we still want to execute our after processing, and then
        // rethrow the problem after that.
        problem = t;
        t.printStackTrace();
    }
}
```

```
doAfterProcessing(request, response);
out.print("<br> after processing bye from filter");
// If there was a problem, we want to rethrow it if it is
// a known type, otherwise log it.
if (problem != null) {
    if (problem instanceof ServletException) {
        throw (ServletException) problem;
    }
    if (problem instanceof IOException) {
        throw (IOException) problem;
    }
    sendProcessingError(problem, response);
}
}

/**
 * Return the filter configuration object for this filter.
 */
public FilterConfig getFilterConfig() {
    return (this.filterConfig);
}

/**
 * Set the filter configuration object for this filter.
 *
 * @param filterConfig The filter configuration object
 */
public void setFilterConfig(FilterConfig filterConfig) {
    this.filterConfig = filterConfig;
}

/**
 * Destroy method for this filter
 */
public void destroy() {
}

/**
 * Init method for this filter
 */
public void init(FilterConfig filterConfig) {
    this.filterConfig = filterConfig;
    if (filterConfig != null) {
        if (debug) {
            log("MyFilter:Initializing filter");
        }
    }
}
```



Advance JAVA (2160707)

```
}

/**
 * Return a String representation of this object.
 */
@Override
public String toString() {
    if (filterConfig == null) {
        return ("MyFilter()");
    }
    StringBuffer sb = new StringBuffer("MyFilter()");
    sb.append(filterConfig);
    sb.append(")");
    return (sb.toString());
}

private void sendProcessingError(Throwable t, ServletResponse response) {
    String stackTrace = getStackTrace(t);

    if (stackTrace != null && !stackTrace.equals("")) {
        try {
            response.setContentType("text/html");
            PrintStream ps = new PrintStream(response.getOutputStream());
            PrintWriter pw = new PrintWriter(ps);
            pw.print("<html>\n<head>\n<title>Error</title>\n</head>\n<body>\n"); //NOI18N

            // PENDING! Localize this for next official release
            pw.print("<h1>The resource did not process correctly</h1>\n<pre>\n");
            pw.print(stackTrace);
            pw.print("</pre></body>\n</html>"); //NOI18N
            pw.close();
            ps.close();
            response.getOutputStream().close();
        } catch (Exception ex) {
        }
    } else {
        try {
            PrintStream ps = new PrintStream(response.getOutputStream());
            t.printStackTrace(ps);
            ps.close();
            response.getOutputStream().close();
        } catch (Exception ex) {
        }
    }
}

public static String getStackTrace(Throwable t) {
```



Advance JAVA (2160707)

```
String stackTrace = null;
try {
    StringWriter sw = new StringWriter();
    PrintWriter pw = new PrintWriter(sw);
    t.printStackTrace(pw);
    pw.close();
    sw.close();
    stackTrace = sw.getBuffer().toString();
} catch (Exception ex) {
}
return stackTrace;
}

public void log(String msg) {
    filterConfig.getServletContext().log(msg);
}
}
```

MyServlet.java

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 *
 * @author faraahdabhoiwala
 */
public class MyServlet extends HttpServlet {

    /**
     * Processes requests for both HTTP GET and POST methods.
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
    }
}
```



Advance JAVA (2160707)

```
PrintWriter out = response.getWriter();
out.print("<br>hello from servlet");
}
```

// <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on the left to edit the code.">

```
/**
 * Handles the HTTP <code>GET</code> method.
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    processRequest(request, response);
}

/**
 * Handles the HTTP <code>POST</code> method.
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    processRequest(request, response);
}

/**
 * Returns a short description of the servlet.
 * @return a String containing servlet description
 */
@Override
public String getServletInfo() {
    return "Short description";
} // </editor-fold>
}
```

WEB-INF

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




Advance JAVA (2160707)

```
<web-app version="3.0" xmlns="http://java.sun.com/xml/ns/javaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-
app_3_0.xsd">
  <context-param>
    <param-name>admin</param-name>
    <param-value>khyati</param-value>
  </context-param>
  <context-param>
    <param-name>user</param-name>
    <param-value>sunny</param-value>
  </context-param>
  <filter>
    <filter-name>MyFilter</filter-name>
    <filter-class>MyFilter</filter-class>
  </filter>
  <filter>
    <filter-name>StudentFilter</filter-name>
    <filter-class>StudentFilter</filter-class>
  </filter>
  <filter-mapping>
    <filter-name>StudentFilter</filter-name>
    <servlet-name>StudentServlet</servlet-name>
  </filter-mapping>
  <servlet>
    <servlet-name>StudentServlet</servlet-name>
    <servlet-class>StudentServlet</servlet-class>
  </servlet>

  <filter-mapping>
    <filter-name>MyFilter</filter-name>
    <servlet-name>MyServlet</servlet-name>
  </filter-mapping>
  <servlet>
    <servlet-name>RetriveContextParam</servlet-name>
    <servlet-class>RetriveContextParam</servlet-class>
    <init-param>
      <param-name>country</param-name>
      <param-value>india</param-value>
    </init-param>
    <init-param>
      <param-name>state</param-name>
      <param-value>gujarat</param-value>
    </init-param>
  </servlet>
  <servlet>
    <servlet-name>SessionCookies</servlet-name>
```



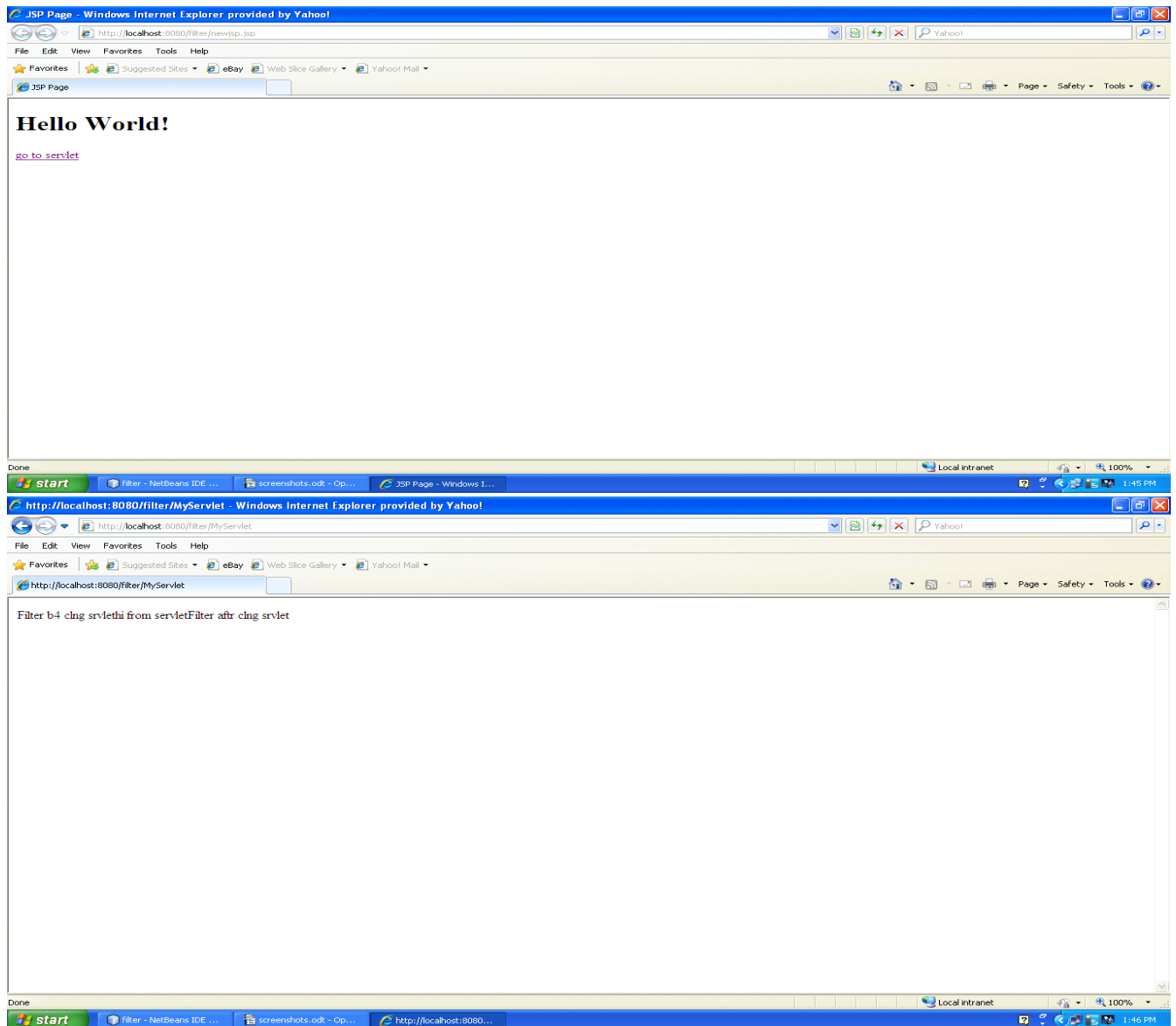
Advance JAVA (2160707)

```
<servlet-class>SessionCookies</servlet-class>
</servlet>
<servlet>
  <servlet-name>testcookie</servlet-name>
  <servlet-class>testcookie</servlet-class>
</servlet>
<servlet>
  <servlet-name>URLRewriting</servlet-name>
  <servlet-class>URLRewriting</servlet-class>
</servlet>
<servlet>
  <servlet-name>JDBCServlet</servlet-name>
  <servlet-class>JDBCServlet</servlet-class>
</servlet>
<servlet>
  <servlet-name>MyServlet</servlet-name>
  <servlet-class>MyServlet</servlet-class>
</servlet>

<servlet-mapping>
  <servlet-name>RetriveContextParam</servlet-name>
  <url-pattern>/RetriveContextParam</url-pattern>
</servlet-mapping>
<servlet-mapping>
  <servlet-name>SessionCookies</servlet-name>
  <url-pattern>/SessionCookies</url-pattern>
</servlet-mapping>
<servlet-mapping>
  <servlet-name>testcookie</servlet-name>
  <url-pattern>/testcookie</url-pattern>
</servlet-mapping>
<servlet-mapping>
  <servlet-name>URLRewriting</servlet-name>
  <url-pattern>/URLRewriting</url-pattern>
</servlet-mapping>
<servlet-mapping>
  <servlet-name>MyServlet</servlet-name>
  <url-pattern>/MyServlet</url-pattern>
</servlet-mapping>
<session-config>
  <session-timeout>30</session-timeout>
</session-config>
</web-app>
```



Advance JAVA (2160707)



Logical Questions:

1. What is the use of filterchain interface?
- 2..which are the applications of filter?



Advance JAVA (2160707)

Practical 10

Aim: Write a program to implement useBean action tag.

Source Code:

JBeanEmpInfo.jsp

```
<%--
    Document : JBeanAddEmpInfo
    Created on : Feb 25, 2016, 8:59:56 AM
    Author : faraahdabhoiwala
--%>

<% @page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body>

        <form method="post" action="JBeanViewEmpDetails.jsp">
            Employee Code :<input type="text" name="empCode">    </input>

            Employee Name :<input type="text" name="empName">    </input>

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

            </input>
        </form>

    </body>
</html>
```

JBeanEmpDetails.jsp

```
<%--
    Document : JBeanViewEmpDetails
    Created on : Feb 25, 2016, 9:08:06 AM
    Author : faraahdabhoiwala
--%>

<% @page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
```



Advance JAVA (2160707)

```
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>JSP Page</title>
</head>
<body>

    <jsp:useBean id="emp" class="p1.JBean" />
    <jsp:setProperty name="emp" property="*" />

    <jsp:getProperty name="emp" property="empCode" />
    <jsp:getProperty name="emp" property="empName" />
</body>
</html>
```

JBean.java

```
package p1;
import java.io.*;
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
/**
 *
 * @author faraahdabhoiwala
 */
public class JBean implements Serializable {
    private String empCode;

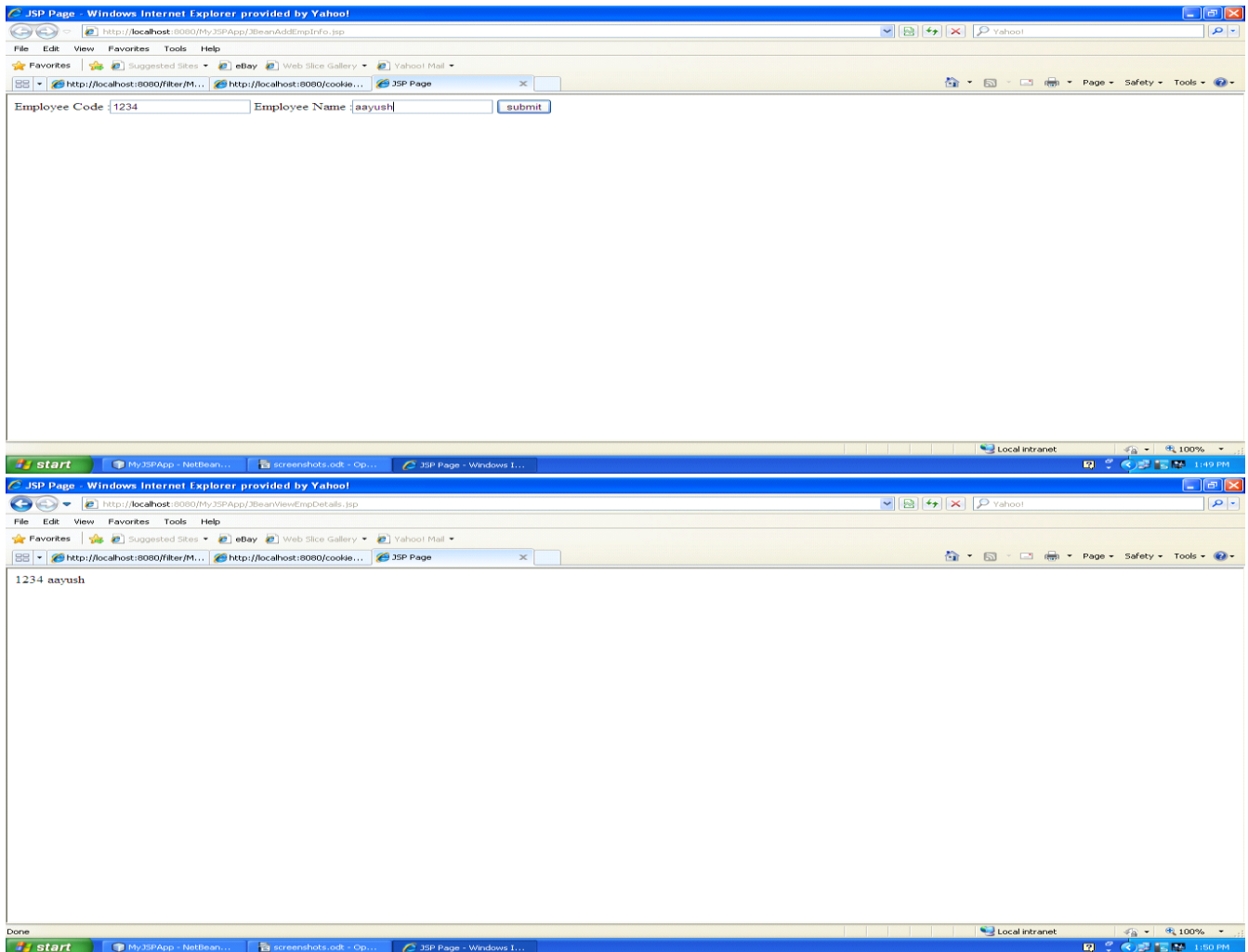
    private String empName;
    public String getEmpName()
    {
        return empName;
    }
    public void setEmpName(String empName)
    {
        this.empName=empName;
    }
    public String getEmpCode()
    {
        return empCode;
    }
    public void setEmpCode(String empCode)
    {
        this.empCode=empCode;
    }
}
```



Advance JAVA (2160707)

}

OUTPUT



Logical Questions:

1. What is the use of Serializable interface?
- 2..which are use os useBean action tag?



Advance JAVA (2160707)

Practical 11

Aim: Write a program to create a Custom tag and use it in a JSP page.

Source Code:

MyTagHandler.java

```
package p1;

import java.io.IOException;
import javax.servlet.jsp.JspWriter;
import javax.servlet.jsp.JspException;
import javax.servlet.jsp.tagext.JspFragment;
import javax.servlet.jsp.tagext.SimpleTagSupport;

public class MyTagHandler extends SimpleTagSupport {
    @Override
    public void doTag() throws JspException, IOException{
        JspWriter out = getJspContext().getOut();
        out.print("hello everyone this is custom tag");
    }
}
```

Mytag_library.tld

```
<?xml version="1.0" encoding="UTF-8"?>
<taglib version="2.1" xmlns="http://java.sun.com/xml/ns/javaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-
jsptaglibrary_2_1.xsd">
    <tlib-version>1.0</tlib-version>
    <short-name>mytag_library</short-name>
    <uri>/WEB-INF/tlds/Mytag_library</uri>

    <tag>
        <name>Display</name>
        <tag-class>p1.MyTagHandler</tag-class>
        <body-content>scriptless</body-content>
    </tag>
</taglib>
```

CustomTag.java

```
<% @taglib prefix="f" uri="/WEB-INF/tlds/Mytag_library" %>
<% @page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
```



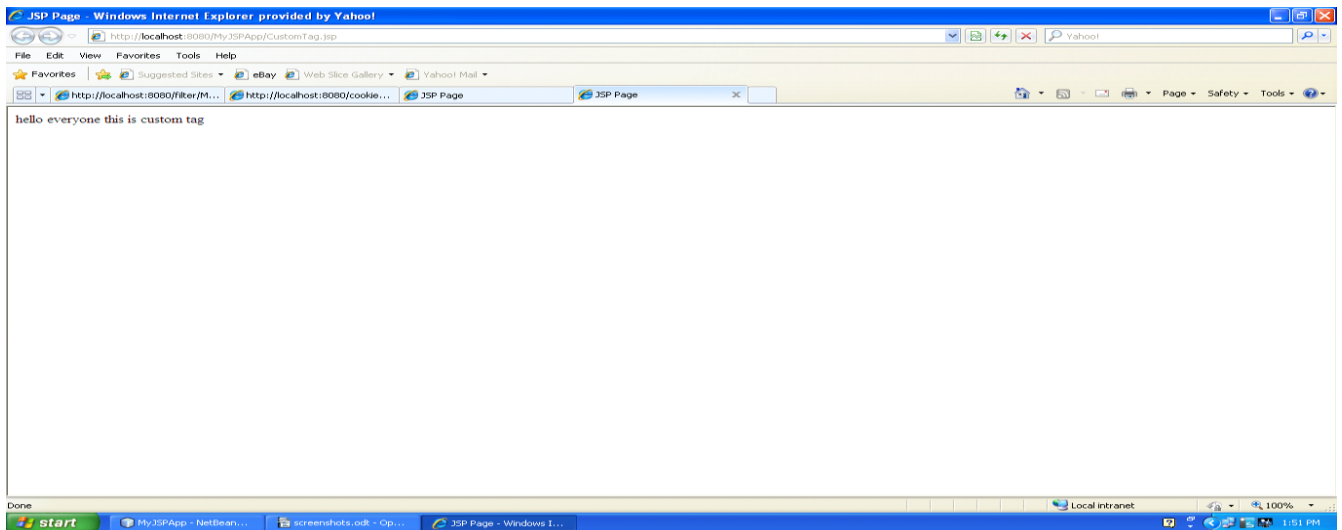
Advance JAVA (2160707)

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>JSP Page</title>
  </head>
  <body>
    <f:Display></f:Display>

  </body>

</html>
```

OUTPUT



Logical Questions:

- 1. Who catch the exceptions and which type of exceptions?*
- 2. Which are the use of custom tag?*



Advance JAVA (2160707)

Practical 12

Aim: Write a program using connectionless protocol in which the client sends a string to server and server converts the string into uppercase and sends it back to client.

Source Code:

DatagramServer.java

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package mynetworkapp;

import java.net.*;
import java.io.*;

/**
 *
 * @author faraahdabhoiwala
 */
public class DatagramServer {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        try{
            DatagramSocket ds=new DatagramSocket(2000);//port no of socket--Socket Exception

            byte b[]=new byte[100];
            DatagramPacket dp=new DatagramPacket(b,b.length);//blank packet
            ds.receive(dp);
            b=dp.getData();//unpack the datapacket//this statment throws IOException

            String d=new String(b).trim();//convert to string
            System.out.println(d);

            //task to be performed on string//

            String d1=d.toUpperCase();//task
```



Advance JAVA (2160707)

```
System.out.println(d1);

//send the string back to the client

byte b1[]=new byte[100]; //byte array for sending data
b1=d1.getBytes(); //convert string form data to bytes
DatagramPacket sp=new DatagramPacket(b1,b1.length,dp.getAddress(),dp.getPort());
ds.send(sp); //send the packet to client--throws IOException

} catch (Exception e) {

    System.out.println(e);
}

}
}
```

DatagramClient.java

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package mynetworkapp;

import java.net.*;
import java.util.Scanner;

/**
 *
 * @author faraahdabhoiwala
 */
public class DatagramClient {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        try {

            DatagramSocket ds=new DatagramSocket(); //throws SocketException

            Scanner sc=new Scanner(System.in);
            System.out.print("Enter a String :");
```



Advance JAVA (2160707)

```
String str=sc.next();

byte b[]=new byte[100];
b=str.getBytes();
//create a packet and store the data in it
DatagramPacket dp=new DatagramPacket(b,b.length,InetAddress.getLocalHost(),2000);
//send the packet
ds.send(dp);

byte b1[]=new byte[100];
//create an empty packet to receive the data
DatagramPacket rp=new DatagramPacket(b1,b1.length);
//fill the packet
ds.receive(rp);
//unpack the data in byte form
b1=rp.getData();
String d=new String(b1).trim();
System.out.println(d);

} catch (Exception e) {

    System.out.println(e);
}

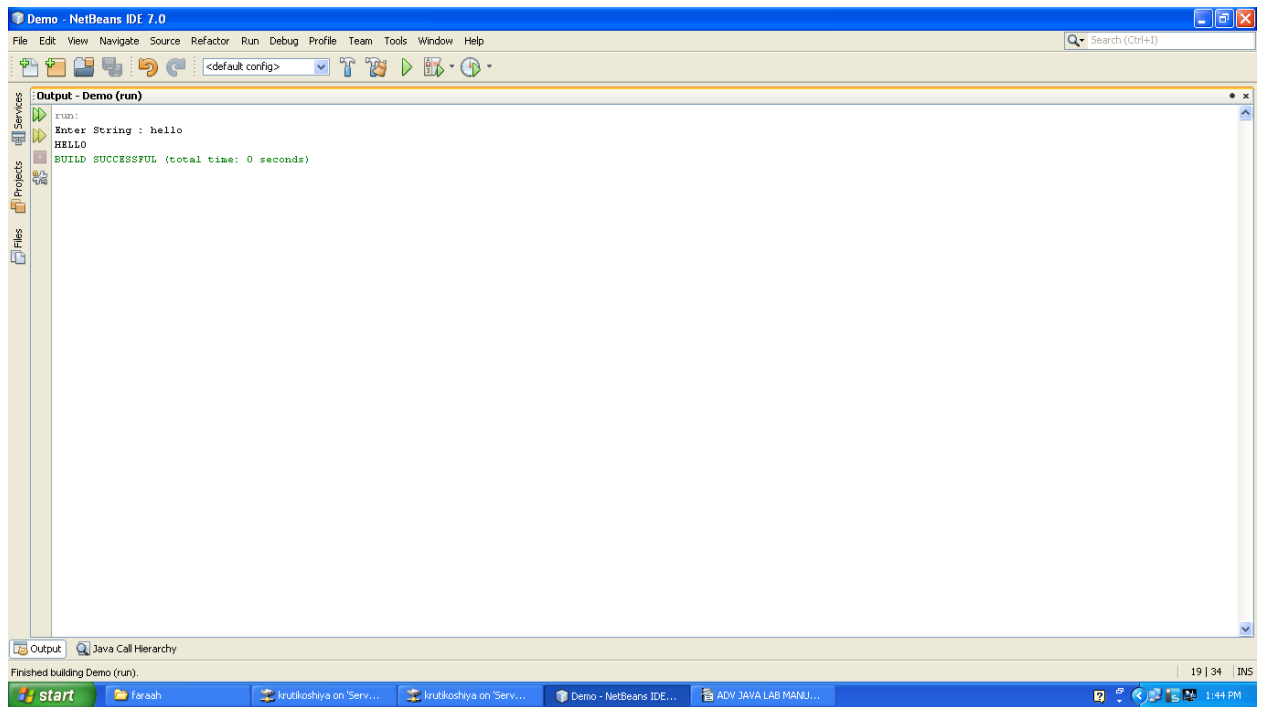
}
```

Logical Questions:

- 1.What is connectionless protocol?
- 2.Name of connectionless protocol.



Advance JAVA (2160707)





Advance JAVA (2160707)

Practical 13

Aim: Write a JSF application to create a login form.

Source Code:

LoginBean.java

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

/*
 * 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.
 */

import java.io.Serializable;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import javax.faces.bean.ManagedBean;
import javax.faces.bean.SessionScoped;

/**
 *
 * @author user
 */
@ManagedBean(name = "loginBean")
//@Dependent
@SessionScoped
public class LoginBean implements Serializable{
    String userName;
    String password;

    public String getPassword() {
        return password;
    }

    public void setPassword(String password) {
        this.password = password;
    }
}
```



Advance JAVA (2160707)

```
public String getUserName() {
    return userName;
}

public void setUserName(String userName) {
    this.userName = userName;
}

public String checkValidUser()
{
    try
    {
        Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

        Connection
con=DriverManager.getConnection("jdbc:odbc:TempDataSource","system","system");
        Statement st=con.createStatement();

        ResultSet rs=st.executeQuery("select * from user1 where username='"+userName+"' and
password='"+password+"'");
        rs.next();

        if(rs.getRow()==1)
        {
            return "success";
        }
        else
        {
            return "failure";
        }

    }catch(Exception e)
    {

        return e.toString();
    }

}

/*if(userName.equals("faraah") && password.equals("faraah"))
return "success";
else
return "failure";*/

/**
```



Advance JAVA (2160707)

```
* Creates a new instance of LoginBean
*/
public LoginBean() {
}

}
/*
import javax.faces.bean.ManagedBean;
import javax.faces.bean.RequestScoped;

@ManagedBean
@RequestScoped
public class LoginBean {

    public LoginBean() {
    }
}
*/
```

login.jsp

```
<%--
    Document   : login
    Created on : Mar 15, 2016, 9:33:42 AM
    Author    : faraahdabhoiwala
--%>
<% @taglib uri="http://java.sun.com/jsf/core" prefix="f"%>
<% @taglib uri="http://java.sun.com/jsf/html" prefix="h"%>

<% @page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<f:view>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <h1>Hello World!</h1>
    <body>
        <h:form>
            Enter UserName :<h:inputText id="userName" value="#{loginBean.userName}"/><br/>

            </h:inputText>
            Enter Password : <h:inputText id="password" value="#{loginBean.password}"/><br/>
            <h:commandButton value="login" action="#{loginBean.checkValidUser()}" />

        </h:form>
```



Advance JAVA (2160707)

```
</body>
</html>
</f:view>
```

success.jsp

```
<%--
    Document   : success
    Created on : Mar 15, 2016, 9:21:09 AM
    Author    : faraahdabhoiwala
--%>

<% @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>login successful</h1>
</body>
</html>
```

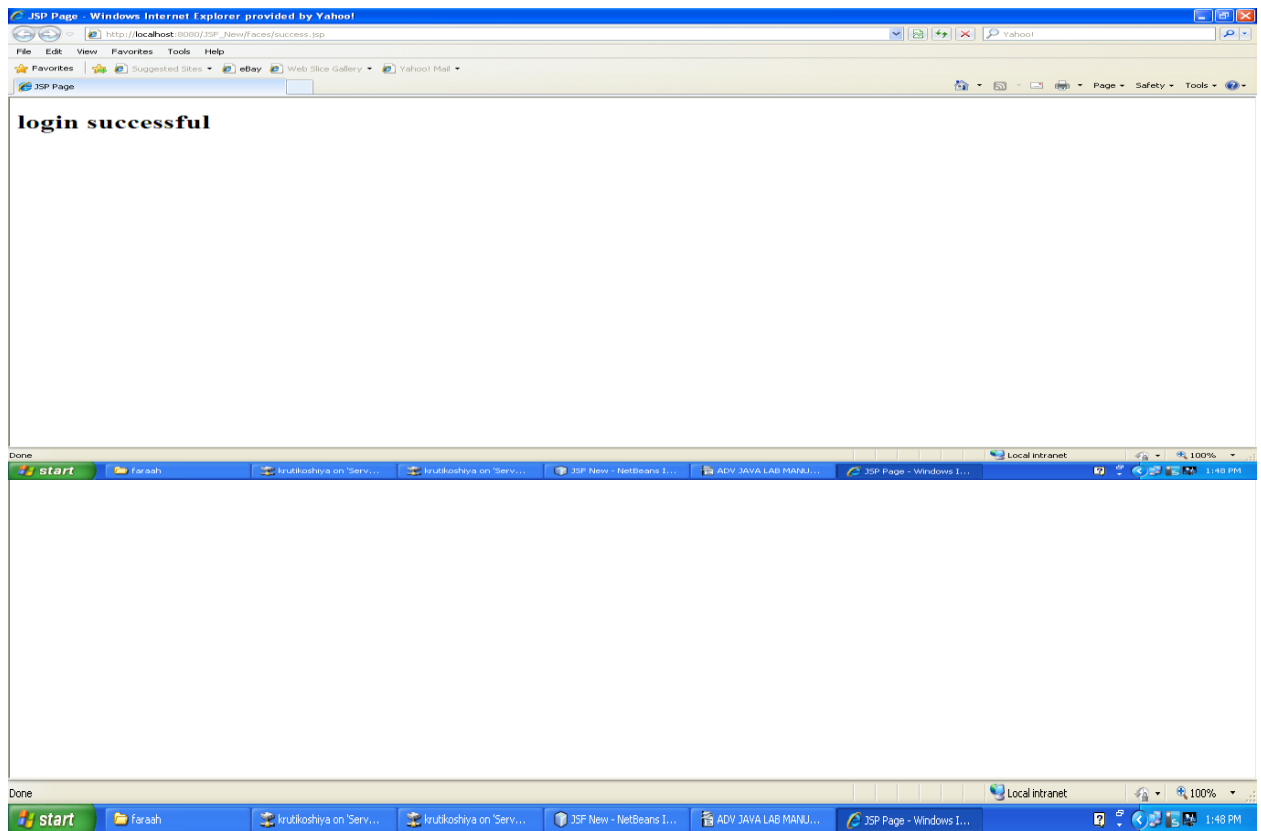
failure.jsp

```
<%--
    Document   : failure
    Created on : Mar 15, 2016, 9:21:21 AM
    Author    : faraahdabhoiwala
--%>

<% @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>invalid username/pwd <br/>login failed</h1>
</body>
</html>
```




Advance JAVA (2160707)



Logical Questions:

1. What is the use of JSF?
2. Which libraries required for JSF?



Advance JAVA (2160707)

Practical 14

Aim: Write a JSF application to insert, update and delete record from database.

Source Code:

LoginBean.java

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

/*
 * 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.
 */

import java.io.Serializable;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import javax.faces.bean.ManagedBean;
import javax.faces.bean.SessionScoped;

/**
 *
 * @author user
 */
@ManagedBean(name = "loginBean")
//@Dependent
@SessionScoped
public class LoginBean implements Serializable{
    String userName;
    String password;

    public String getPassword() {
        return password;
    }

    public void setPassword(String password) {
        this.password = password;
    }
}
```



Advance JAVA (2160707)

```
}

public String getUsername() {
    return userName;
}

public void setUsername(String userName) {
    this.userName = userName;
}

public String checkValidUser()
{
    try
    {
        Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

        Connection
con=DriverManager.getConnection("jdbc:odbc:TempDataSource","system","system");
        Statement st=con.createStatement();

        ResultSet rs=st.executeQuery("select * from user1 where username='"+userName+"' and
password='"+password+"'");
        rs.next();

        if(rs.getRow()==1)
        {
            return "success";
        }
        else
        {
            return "failure";
        }
    }catch(Exception e)
    {

        return e.toString();
    }

}

/*if(userName.equals("faraah") && password.equals("faraah"))
return "success";
else
return "failure";*/
```



Advance JAVA (2160707)

```
/**
 * Creates a new instance of LoginBean
 */
public LoginBean() {
}

}
/*
import javax.faces.bean.ManagedBean;
import javax.faces.bean.RequestScoped;

@ManagedBean
@RequestScoped
public class LoginBean {

    public LoginBean() {
    }
}
*/
```

login.jsp

```
<%--
    Document : login
    Created on : Mar 15, 2016, 9:33:42 AM
    Author : faraahdabhoiwala
--%>
<% @taglib uri="http://java.sun.com/jsf/core" prefix="f"%>
<% @taglib uri="http://java.sun.com/jsf/html" prefix="h"%>

<% @page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<f:view>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <h1>Hello World!</h1>
    <body>
        <h:form>
            Enter UserName :<h:inputText id="userName" value="#{loginBean.userName}"><br/>

            </h:inputText>
            Enter Password : <h:inputText id="password" value="#{loginBean.password}"/><br/>
            <h:commandButton value="login" action="#{loginBean.checkValidUser()}" />
        </h:form>
    </body>
</html>
```



Advance JAVA (2160707)

</h:form>

</body>

</html>

</f:view>

success.jsp

<%--

Document : success

Created on : Mar 15, 2016, 9:21:09 AM

Author : faraahdabhoiwala

--%>

<% @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>login successful</h1>

</body>

</html>

failure.jsp

<%--

Document : failure

Created on : Mar 15, 2016, 9:21:21 AM

Author : faraahdabhoiwala

--%>

<% @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>

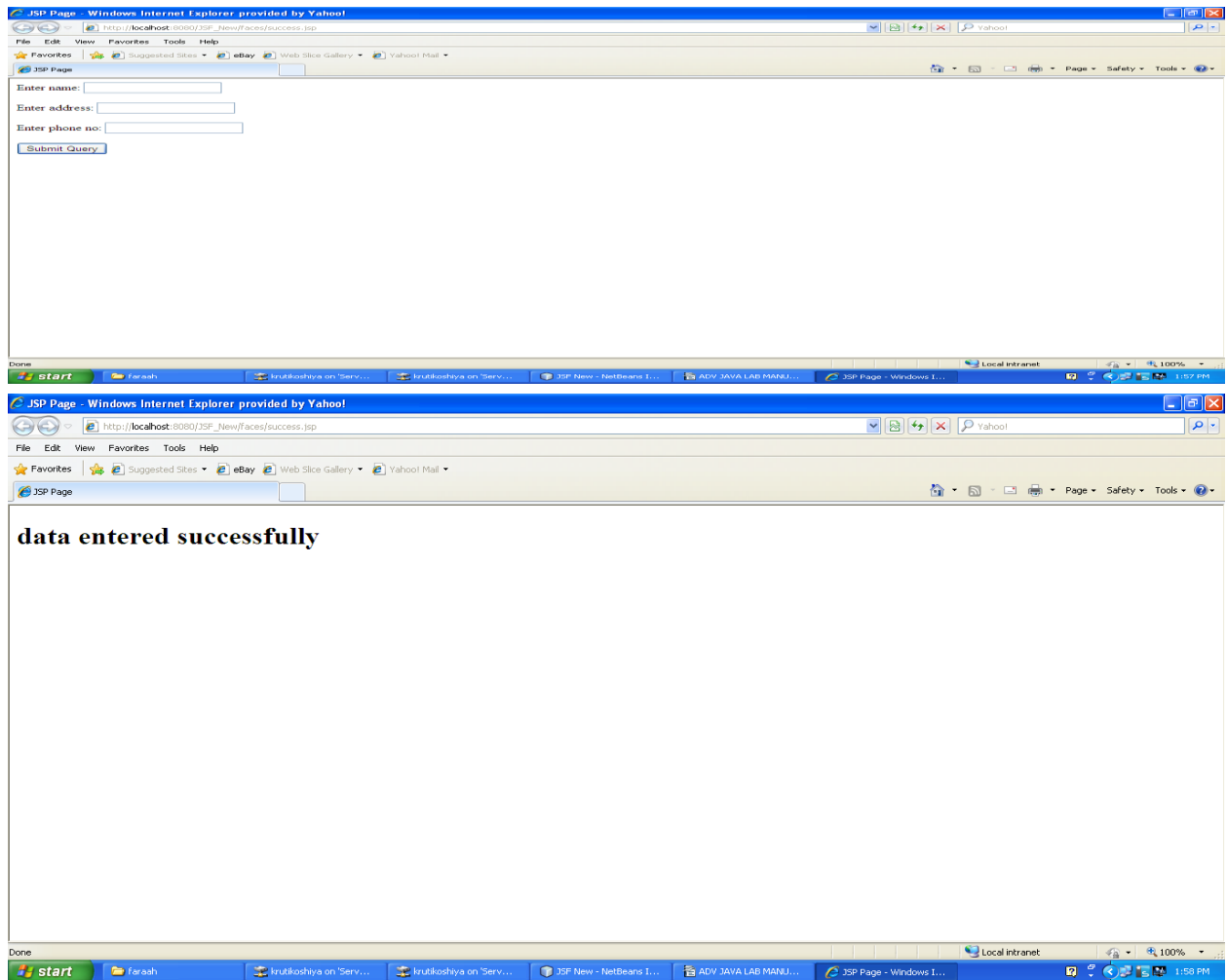
<h1>invalid username/pwd
login failed</h1>

</body>

</html>



Advance JAVA (2160707)



Logical Questions:

- 1.How JSF application connect with Database?*
- 2.How data retrieve from the database?*