

SIR GURUDAS MAHAVIDYALAYA

Department of Computer Sc.



LAB MANUAL

Subject: Advance Java Programming

Paper Code: DSE B4 (Practical)

Advanced Java Programming Lab (7MCE1P1)

1. Write an Applet which will play two sound notes in a sequence continuously use the play () methods available in the applet class and the methods in the Audio clip interface.
2. Create a Japplet using swing control, which will create the layout shown below and handle necessary events.

FORMAT

Enter your Name:	
Enter your Age:	
Select your s/w: * Oracle *Visual Basic	
OK	Cancel

3. Use JDBC connectivity and create Table, insert and update data.
4. Write a program in Java to implement a Client/Server application using RMI.
5. Write a program in Java to create a Cookie and set the expiry time of the same.
6. Write a program in Java to create Servlet to count the number of visitors to a web page.
7. Write a program in Java to create a form and validate a password using Servlet.
8. Develop a Java Bean to demonstrate the use of the same.
9. Write a program in Java to convert an image in RGB to a Grayscale image.
10. Develop Chat Server using Java.

EX.NO:1

PLAY TWO AUDIOS IN A SEQUENCE CONTINUOUSLY USING AUDIOCLIP INTERFACE

Aim:

To write a java applet program to play two sound notes simultaneously using the play() method in AudioClip interface.

Algorithm:

Step 1:

Start the program.

Step 2:

Import java packages such as java.applet.*, java.awt.*and java.awt.event.*.

Step 3:

Define class with name 'pgm3' and extends it from the class 'Applet' and also implements the interface 'ActionListener'.

Step 4:

Define the init() method and create two button objects labeled as 'play audio 1','paly audio 2' respectively.

Step 5:

Add the buttons to panel and addActionlistener for each button to handle ActionEvent.

Step 6:

Define actionPerformed() method for handling click events of buttons.

Step 7:

Stop the program.

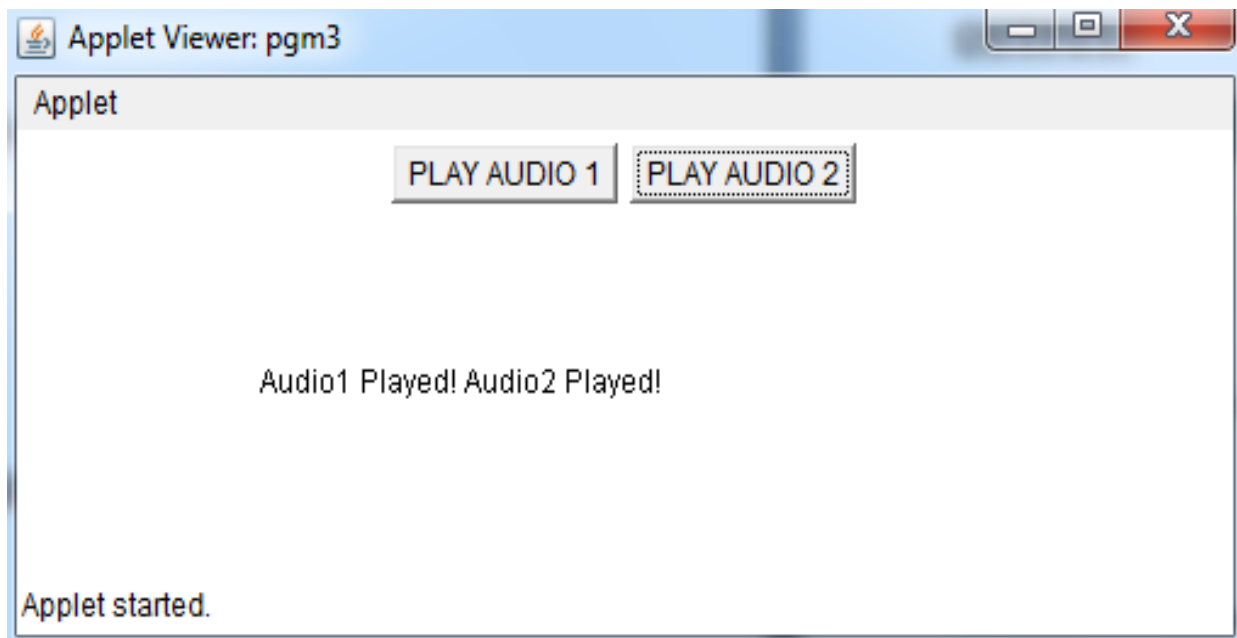
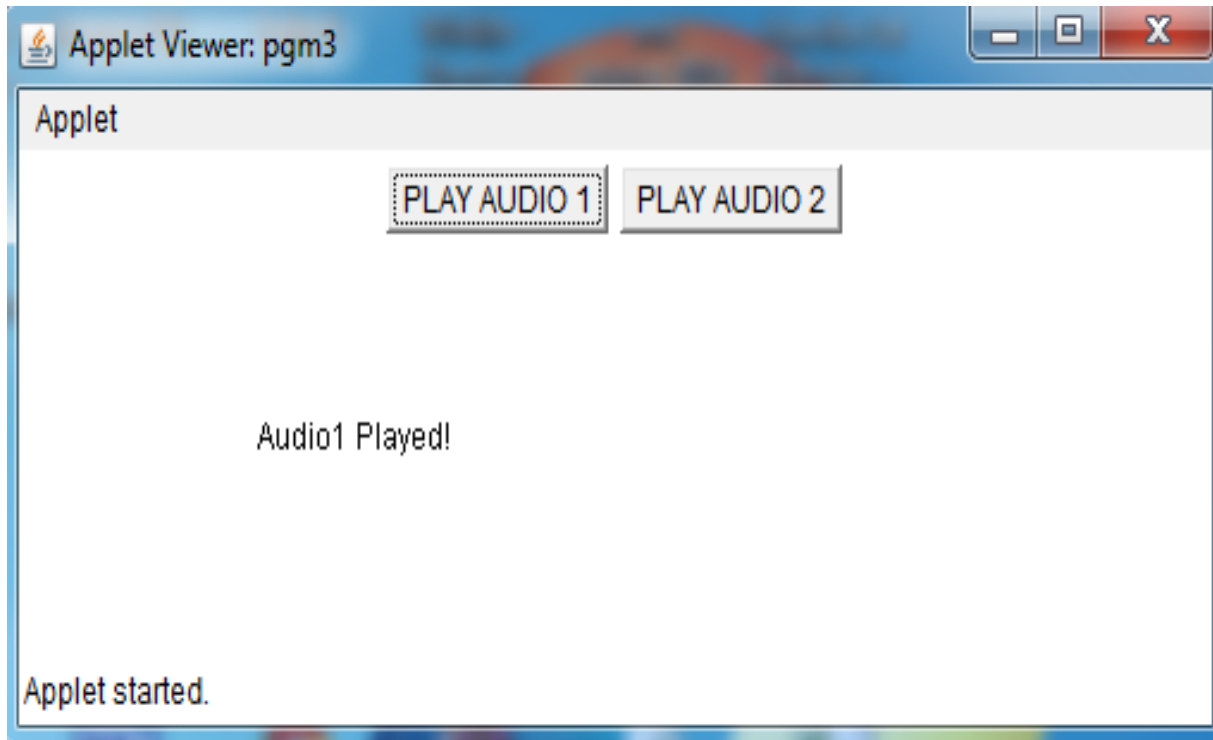
SOURCE CODE:

// Play Two Audios in a Sequence Continuously Using AudioClip Interface

```
import java.applet.*;
import java.awt.*;
import java.awt.event.*;
/*<applet code="pgm3" height=500 width=500></applet>*/
public class pgm3 extends Applet implements ActionListener
{
    Button b1,b2;
    AudioClip ac;
    String str="";
    public void init()
    {
        b1=new Button("PLAY AUDIO 1");
        b2=new Button("PLAY AUDIO 2");
        add(b1);
        add(b2);
        b1.addActionListener(this);
        b2.addActionListener(this);
    }
    public void actionPerformed(ActionEvent ae)
    {
        if(ae.getSource()==b1)
        {
            ac=getAudioClip(getCodeBase(),"x1.wav");
            str += "Audio1";
        }
        else if(ae.getSource()==b2)
        {
            ac=getAudioClip(getCodeBase(),"x2.wav");
            str += " Audio2";
        }
        ac.play();
        str += " Played!";
        repaint();
    }
    public void paint(Graphics g)
    {
        g.drawString(str,100,100);
    }
}
```

OUTPUT:

```
C:\Program Files \Java\jdk1.7.0\bin>javac pgm3.java  
C:\Program Files \Java\jdk1.7.0\bin>appletviewer pgm3.java
```



RESULT:

The above program has been executed successfully and the output was verified.

EX.NO:2

CREATE SAMPLE APPLICATION FORM USING JAPPLET

Aim:

To write a java program to create sample application form in JApplet using swing control.

Algorithm:

Step 1:

Define class pgm8 which extends from JApplet and implement the interface ActionListener.

Step 2:

Create object for JLabel, JTextfield, JButton, JCheckbox, JRadiobutton as needed.

Step 3:

Add all the components into the container.

Step 4:

Add Action listener to the buttons for handling events.

Step 5:

Define actionPerformed() method, write a source code for button "ok" and "cancel", Print given information while clicking "ok" button. If we click "cancel" button reset the form.

Step 6:

Stop the program.

SOURCE CODE:

```
//Create Sample Application Form Using JApplet

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
/*<applet code="pgm8" height=500 width=700></applet>*/
public class pgm8 extends JApplet implements ActionListener
{
    JButton b1,b2;
    JTextField t1,t2;
    JLabel l1,l2,l3,l4,msg;
    Container cp;
    JRadioButton r1,r2,r3;
    JCheckBox ch1,ch2,ch3;
    String str,x1,x2;
    ButtonGroup bg,bg1;
    JPanel p1,p2,p3,p4,p5,p6;
    public void init()
    {
        cp=getContentPane();
        cp.setLayout(new GridLayout(7,1));
        p1=new JPanel();
        p1.setLayout(new FlowLayout(FlowLayout.CENTER,10,10));
        p2=new JPanel();
        p2.setLayout(new FlowLayout(FlowLayout.CENTER,10,10));
        p3=new JPanel();
        p3.setLayout(new FlowLayout(FlowLayout.CENTER,10,10));
        p4=new JPanel();
        p4.setLayout(new FlowLayout(FlowLayout.CENTER,10,10));
        p5=new JPanel();
        p5.setLayout(new FlowLayout(FlowLayout.CENTER,10,10));
        p6=new JPanel();
        p6.setLayout(new FlowLayout(FlowLayout.CENTER,10,10));
        cp.add(p1);
        cp.add(p2);
        cp.add(p3);
        cp.add(p4);
        cp.add(p5);
        cp.add(p6);
        l1=new JLabel("Enter your Name");
        t1=new JTextField(20);
        l2=new JLabel("Enter your Age");
        t2=new JTextField(20);
        bg=new ButtonGroup();
        l3=new JLabel("Enter your City");
        r1=new JRadioButton("Madurai");
        r2=new JRadioButton("Chennai");
        r3=new JRadioButton("Trichy");
```

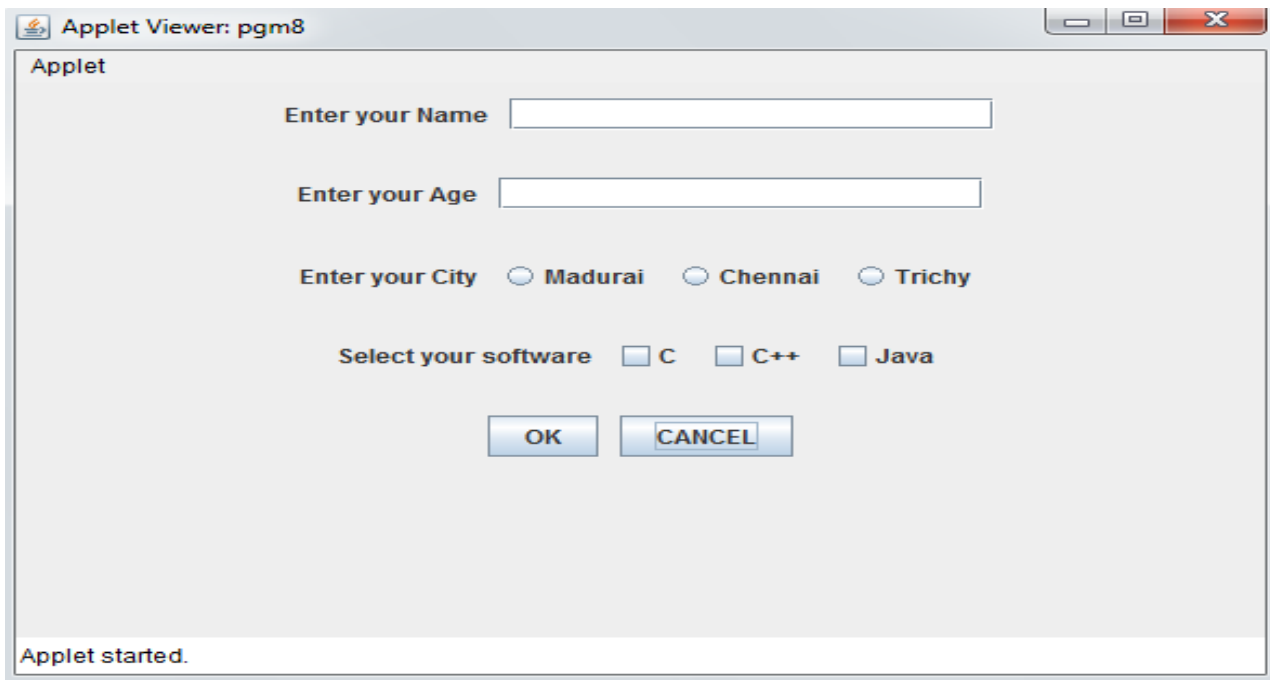
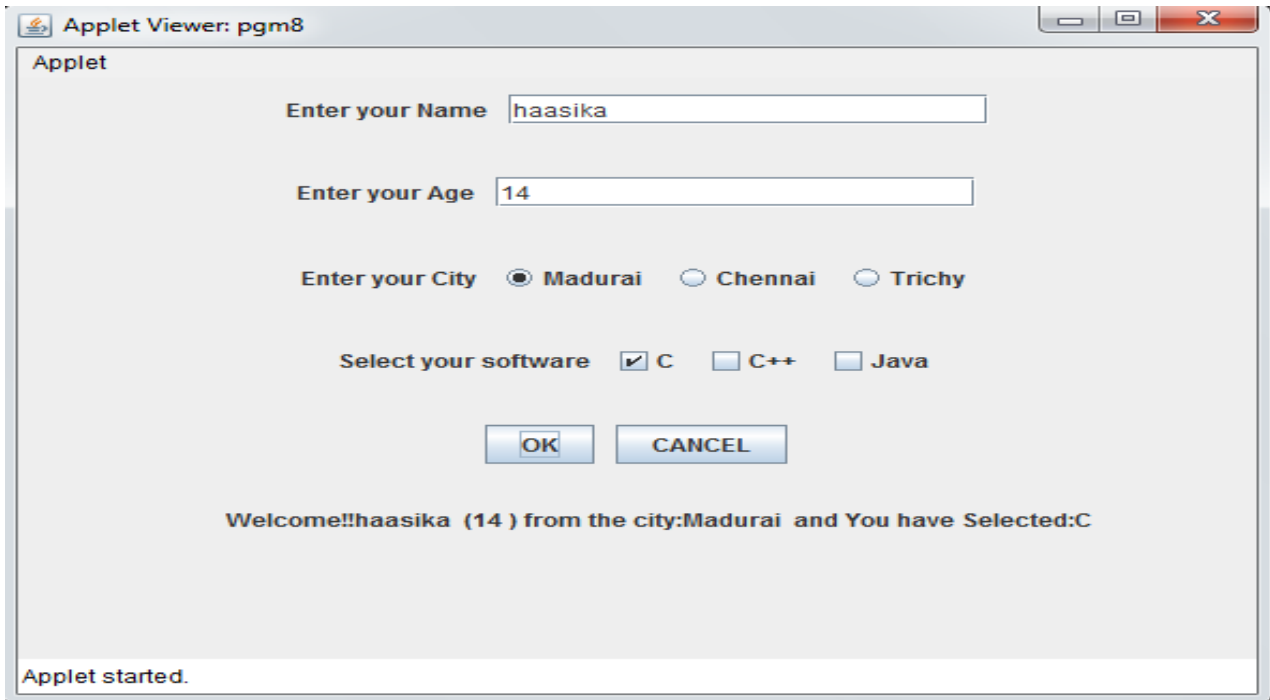
```
bg.add(r1);
bg.add(r2);
bg.add(r3);
l4=new JLabel("Select your software");
ch1=new JCheckBox("C");
ch2=new JCheckBox("C++");
ch3=new JCheckBox("Java");
bg1=new ButtonGroup();
bg1.add(ch1);
bg1.add(ch2);
bg1.add(ch3);
b1=new JButton("OK");
b2=new JButton("CANCEL");
b1.addActionListener(this);
b2.addActionListener(this);
r1.addActionListener(this);
r2.addActionListener(this);
r3.addActionListener(this);
ch1.addActionListener(this);
ch2.addActionListener(this);
ch3.addActionListener(this);
msg=new JLabel("");
p6.add(msg);
p1.add(l1);
p1.add(t1);
p2.add(l2);
p2.add(t2);
p3.add(l3);
p3.add(r1);
p3.add(r2);
p3.add(r3);
p4.add(l4);
p4.add(ch1);
p4.add(ch2);
p4.add(ch3);
p5.add(b1);
p5.add(b2);
msg=new JLabel("");
p6.add(msg);
}
public void actionPerformed(ActionEvent ae)
{
    if(ae.getSource()==b1)
    {
        if(r1.isSelected()==true)
        {
            x1=r1.getText();
        }
        else if(r2.isSelected()==true)
        {
            x1=r2.getText();
        }
    }
}
```



```
    }
    else if(r3.isSelected()==true)
    {
        x1=r3.getText();
    }
    str=" from the city:"+x1;
    if(ch1.isSelected()==true)
    {
        x2=ch1.getText();
    }
    if(ch2.isSelected()==true)
    {
        x2+=","+ch2.getText();
    }
    if(ch3.isSelected()==true)
    {
        x2+=","+ch3.getText();
    }
    str+=" and You have Selected:"+x2;
    msg.setText("Welcome!!"+t1.getText()+" ("+t2.getText()+")"+str);
}
if(ae.getSource()==b2)
{
    t1.setText("");
    t2.setText("");
    ch1.setSelected(false);
    ch2.setSelected(false);
    ch3.setSelected(false);
    r1.setSelected(false);
    r2.setSelected(false);
    r3.setSelected(false);
    msg.setText("Your Registration is Cancelled");
}
}
}
```

OUTPUT:

```
C:\Program Files \Java\jdk1.7.0\bin>javac pgm8.java  
C:\Program Files \Java\jdk1.7.0\bin>appletviewer pgm8.java
```



RESULT:

The above program has been executed successfully and the output was verified.

EX.NO:3

USE JDBC CONNECTIVITY AND CREATE TABLE, INSERT, DELETE AND UPDATE DATA

Aim:

To write a java program using JDBC Connection with ODBC technique to create table and perform insert, update and delete data using MS-Access.

Way to Procedure:

Database Creation:

Step 1:

Go to Start→Programs→Microsoft Access 2007→Select the option BLANK DATABASE

Step 2:

Choose Path of database and give filename “student” with file format Microsoft access 2002-2003 format (.mdb).

Step 3:

Save table as “student_tab”.

Step 4:

Select design view of table by clicking right button and give fields of table such as id, name with respective data type number, text respectively and set unique key in id.

Step 5:

Give sample records and save the database and table data.

Step 6:

Close the package Microsoft access.

Data Source name creation:

Step 1:

Go to → start→control panel→administrative tools→data source (ODBC) and get the wizard.

Step 2:

In DSN wizard, click add button and choose driver as “Microsoft diver do access (*.mdb).

Step 3:

Give data source name as “stud” and select path of database then click OK.

Step 4:

Exit from ODBC wizard.

Algorithm:

Step1:

Start the program.

Step 2:

Include packages java.io and java.sql.

Step 3:

Define class with name “jdbc” and define the main function.

Advanced Java Programming Lab

Step 4:

Declare objects for Connection, Statement, ResultSet and also declare the object for BufferedReader class.

Step 5:

Declare local variables ch,rno,n as integer and na as String.

Step 6:

Register the JdbcOdbcDriver and make a connection using getConnection() by giving Data Source Name "Stud".

Step 7:

Define switch case 1 for insert records,case 2 for delete records,3 for update records and 4 for Display records.

Step 8:

Do Step 7 until will give choice > 4.

Step 9:

Close Statement object and Connection object.

Step 10:

Stop the Program.

SOURCE CODE:

```
// Use JDBC connectivity and create table, insert, update and delete data

import java.io.*;
import java.sql.*;
class jdbc
{
public static void main(String ar[])throws Exception
{
    Connection con;
    Statement st;
    ResultSet rs;
    BufferedReader br=new BufferedReader (new InputStreamReader(System.in));
    int ch,rno,n;
    String na;
    Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
    con=DriverManager.getConnection("jdbc:odbc:stud");
    st=con.createStatement();
    do
    {
        System.out.println("DATABASE MANIPULATION USING JDBC");
        System.out.println("1.Insert\n2.Delete\n3.Update\n4.Display");
        System.out.println("Enter the choice");
        ch=Integer.parseInt(br.readLine());
        switch(ch)
        {
            case 1:
                System.out.println("Enter Id to Insert:");
                rno=Integer.parseInt(br.readLine());
                System.out.println("Enter name to Insert:");
                na=br.readLine();
                try
                {
                    n=st.executeUpdate("insert into student_tab values("+rno+", '"+na+"'");
                    System.out.println(n+" row Inserted!!");
                }
                catch(SQLException e) { }
                break;
            case 2:
                System.out.println("Enter Id to Delete:");
                rno=Integer.parseInt(br.readLine());
                try
                {
                    n=st.executeUpdate("delete * from student_tab where id="+rno);
                    System.out.println(n+" row Deleted!!");
                }
                catch(SQLException e){}
                break;
            case 3:
                System.out.println("Enter Id to Edit:");
```

Advanced Java Programming Lab

```
rno=Integer.parseInt(br.readLine());
System.out.println("Enter name to Edit:");
na=br.readLine();
try
{
    n=st.executeUpdate("update student_tab set name='"+na+"' where
                                                                id="+rno);

    System.out.println(n+" row Updated!!");
}
catch(SQLException e){}
break;
case 4:
try
{
    rs=st.executeQuery("select * from student_tab");
    System.out.println("ID\tNAME\n*****");
    while(rs.next())
    {
        System.out.println(rs.getInt(1)+"\t"+rs.getString(2));
    }
}
catch(SQLException e) {}
break;
default:
    System.out.println("Invalid Choice");
}
}while(ch<=4);
st.close();
con.close();
}
}
```

OUTPUT:

```
C:\Program Files \Java\jdk1.7.0\bin>javac jdbc.java
C:\Program Files \Java\jdk1.7.0\bin>java jdbc
```

DATABASE MANIPULATION USING JDBC

1.Insert

2.Delete

3.Update

4.Display

Enter Choice:

1

Enter Id to Insert:

111

Enter name to Insert:

haafi

1 row Inserted!!

DATABASE MANIPULATION USING JDBC

1.Insert

2.Delete

3.Update

4.Display

Enter Choice:4

ID NAME

111 haafi

222 sita

DATABASE MANIPULATION USING JDBC

1.Insert

2.Delete

3.Update

4.Display

Enter Choice:2

Enter Id to Delete:

222

1 row Deleted!!

DATABASE MANIPULATION USING JDBC

1.Insert

2.Delete

3.Update

4.Display

Enter Choice:

4

ID NAME

111 haafi

DATABASE MANIPULATION USING JDBC

1.Insert

2.Delete

3.Update

```
4.Display
Enter Choice:3
Enter Id to Edit:
111
Enter name to Edit:
haasika
1 row Updated!!
DATABASE MANIPULATION USING JDBC
1.Insert
2.Delete
3.Update
4.Display
Enter Choice:
4
ID  NAME
*****
111  haasika
DATABASE MANIPULATION USING JDBC
1.Insert
2.Delete
3.Update
4.Display
Enter Choice: 5
Invalid Choice
```

RESULT:

The above program has been executed successfully and the output was verified.

EX.NO:4

IMPLEMENT A CLIENT/SERVER APPLICATION USING RMI

Aim:

To write a java program to implement a Client/Server application using RMI.

Algorithm:

Program 1: Define the Remote Interface

Step 1:

Start the program.

Step 2:

Import the package java.rmi.*.

Step 3:

Define interface "AddServerIntf" by extends from Remote.

Step 4:

Declare the methods to perform arithmetic operation add, sub, mul, div, modulo and throws RemoteException.

Step 5:

Stop the program.

Program 2: Implement Remote Interface

Step 1:

Start the program.

Step 2:

Import the packages java.rmi.* and java.rmi.server.*.

Step 3:

Define class AddServerImpl by extends from "UnicastRemoteObject" and implements "AddServerIntf".

Step 4:

Define the procedure for interface methods by throwing RemoteException.

Step 5:

Stop the program.

Program 3: Implementation of Server Machine

Step 1:

Start the program.

Step 2:

Import the packages java.rmi.* and java.net.*

Step 3:

Define class server with main function.

Step 4:

Create object for the class AddServerImpl.

Step 5:

Using naming.rebind() method add the interface to the server .

Step 6:

Stop the program.

Program 4: Implementation of Client Machine

Step 1:

Start the program.

Step 2:

Define class "client" with main() function

Step 3:

Create object for server interface with proper URL definition using naming.lookup();

Step 4:

Using this object call required methods and handling exception.

Step 5:

Stop the program.

SOURCE CODE:

// 1. Define the Remote Interface

```
import java.rmi.*;
public interface AddServerIntf extends Remote
{
    int add(int a,int b) throws RemoteException;
    int sub(int a,int b) throws RemoteException;
    int mul(int a,int b) throws RemoteException;
    int div(int a,int b) throws RemoteException;
    int mod(int a,int b) throws RemoteException;
}
```

// 2. Implement Remote Interface

```
import java.rmi.*;
import java.rmi.server.*;
public class AddServerImpl extends UnicastRemoteObject implements AddServerIntf
{
    public AddServerImpl() throws RemoteException
    {}
    public int add(int a,int b)throws RemoteException
    {
        return (a+b);
    }
    public int sub(int a,int b)throws RemoteException
    {
        return (a-b);
    }
    public int mul(int a,int b)throws RemoteException
    {
        return (a*b);
    }
    public int div(int a,int b)throws RemoteException
    {
        return (a/b);
    }
    public int mod(int a,int b)throws RemoteException
    {
        return (a%b);
    }
}
```

// 3. Implementation of Server Machine

```
import java.rmi.*;
import java.net.*;
public class AddServer
{
    public static void main(String args[])
```

```
{
try
{
    AddServerImpl obj = new AddServerImpl();
    Naming.rebind("addserver", obj);
    System.out.println("server started");
}
catch (Exception e)
{
    System.out.println("Exception: " + e);
}
}
```

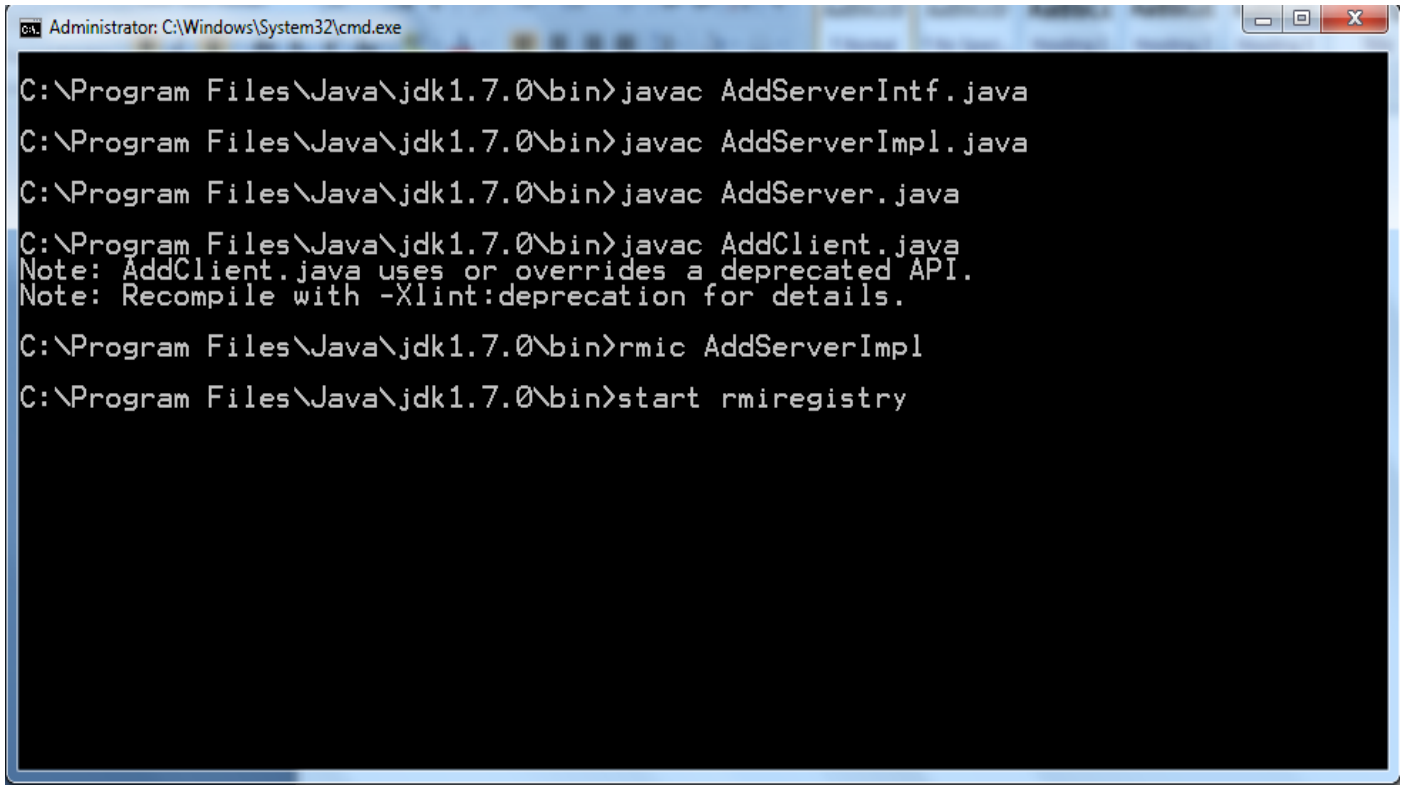
// 4. Implementation of Client Machine

```
import java.rmi.*;
import java.io.*;
public class AddClient

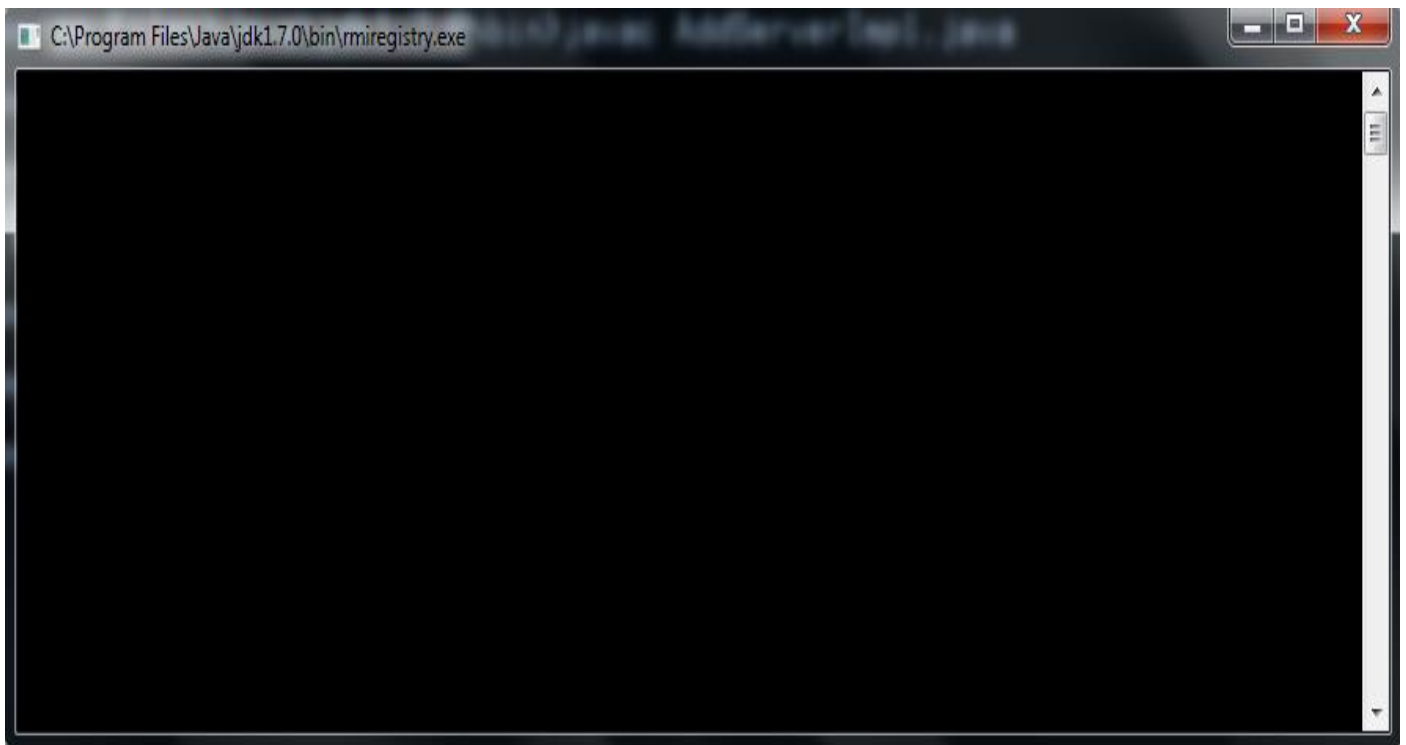
{
    public static void main(String args[])
    {
        try
        {
            DataInputStream ds=new DataInputStream(System.in);
            String s="rmi://MY-PC/addserver";
            AddServerIntf obj = (AddServerIntf)Naming.lookup(s);
            System.out.println("ENTER THE VALUES FOR a & b:");
            int a=Integer.parseInt(ds.readLine());
            int b=Integer.parseInt(ds.readLine());
            System.out.println("ADDITION="+obj.add(a,b));
            System.out.println("SUBTRACTION="+obj.sub(a,b));
            System.out.println("MULTIPLICATION="+obj.mul(a,b));
            System.out.println("DIVISION="+obj.div(a,b));
            System.out.println("MODULODIVISION="+obj.mod(a,b));
        }
        catch (Exception e)
        {
            System.out.println("Exception: " + e);
        }
    }
}
```

OUTPUT:

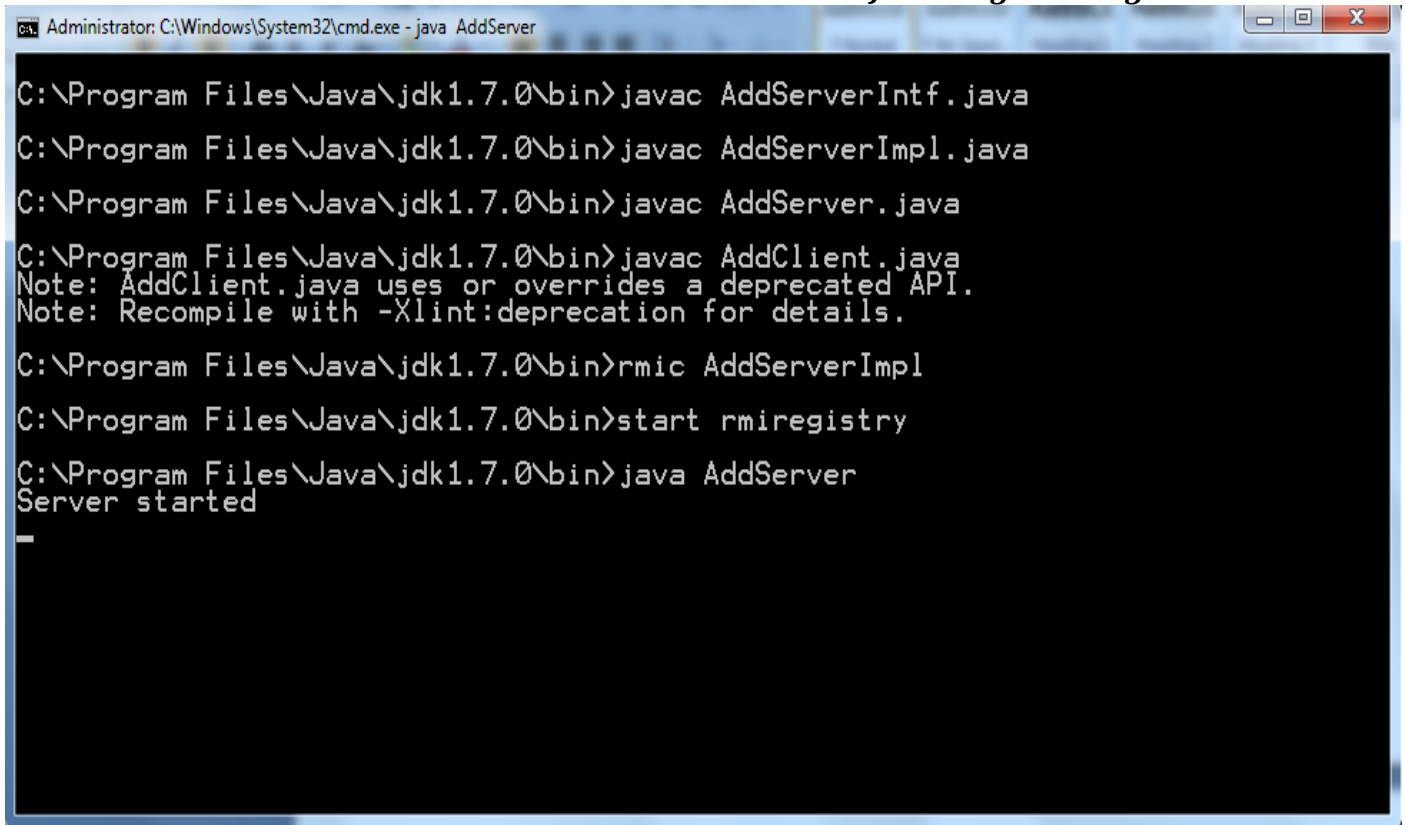
LOCAL HOST SERVER SIDE COMMAND WINDOW:



```
Administrator: C:\Windows\System32\cmd.exe
C:\Program Files\Java\jdk1.7.0\bin>javac AddServerIntf.java
C:\Program Files\Java\jdk1.7.0\bin>javac AddServerImpl.java
C:\Program Files\Java\jdk1.7.0\bin>javac AddServer.java
C:\Program Files\Java\jdk1.7.0\bin>javac AddClient.java
Note: AddClient.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
C:\Program Files\Java\jdk1.7.0\bin>rmic AddServerImpl
C:\Program Files\Java\jdk1.7.0\bin>start rmiregistry
```

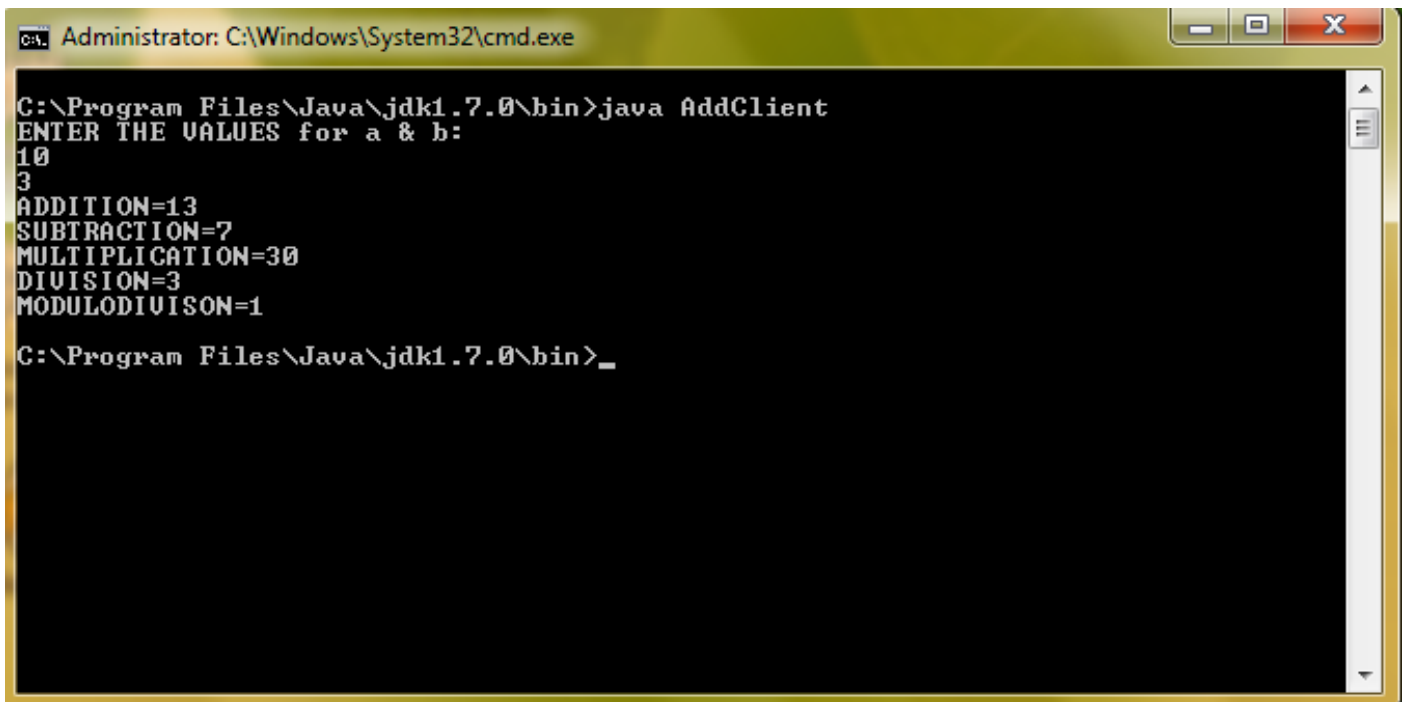


```
C:\Program Files\Java\jdk1.7.0\bin\rmiregistry.exe
```



```
Administrator: C:\Windows\System32\cmd.exe - java AddServer
C:\Program Files\Java\jdk1.7.0\bin>javac AddServerIntf.java
C:\Program Files\Java\jdk1.7.0\bin>javac AddServerImpl.java
C:\Program Files\Java\jdk1.7.0\bin>javac AddServer.java
C:\Program Files\Java\jdk1.7.0\bin>javac AddClient.java
Note: AddClient.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
C:\Program Files\Java\jdk1.7.0\bin>rmic AddServerImpl
C:\Program Files\Java\jdk1.7.0\bin>start rmiregistry
C:\Program Files\Java\jdk1.7.0\bin>java AddServer
Server started
_
```

LOCAL HOST CLIENT SIDE COMMAND WINDOW:



```
Administrator: C:\Windows\System32\cmd.exe
C:\Program Files\Java\jdk1.7.0\bin>java AddClient
ENTER THE VALUES for a & b:
10
3
ADDITION=13
SUBTRACTION=7
MULTIPLICATION=30
DIVISION=3
MODULODIVISION=1
C:\Program Files\Java\jdk1.7.0\bin>_
```

RESULT:

The above program has been executed successfully and the output was verified.

EX.NO:5

CREATE A COOKIE AND SET THE EXPIRY TIME

Aim:

To write a java program to create a cookie and set the expiry time of the same.

Algorithm:

Step 1:

Start the program.

Step 2:

Create html file which contains the text fields for first and last name.

Step 3:

Set submit button also.

Step 4:

Create java program and import needed package.

Step 5:

Define class cook by extending HttpServlet.

Step 6:

Define objects for "cookie" class within Doget () method.

Step 7:

Set expiry time for 2 cookies using SetMaxAge ()

Step 8:

Add cookies to response object.

Step 9:

Set content type of page.

Step 10:

Create object for printWriter for printing firstname and lastname which is from html file.

Step 11:

Stop the program.

SOURCE CODE:

```
// Create a Cookie and Set the Expiry Time
```

Hello.html

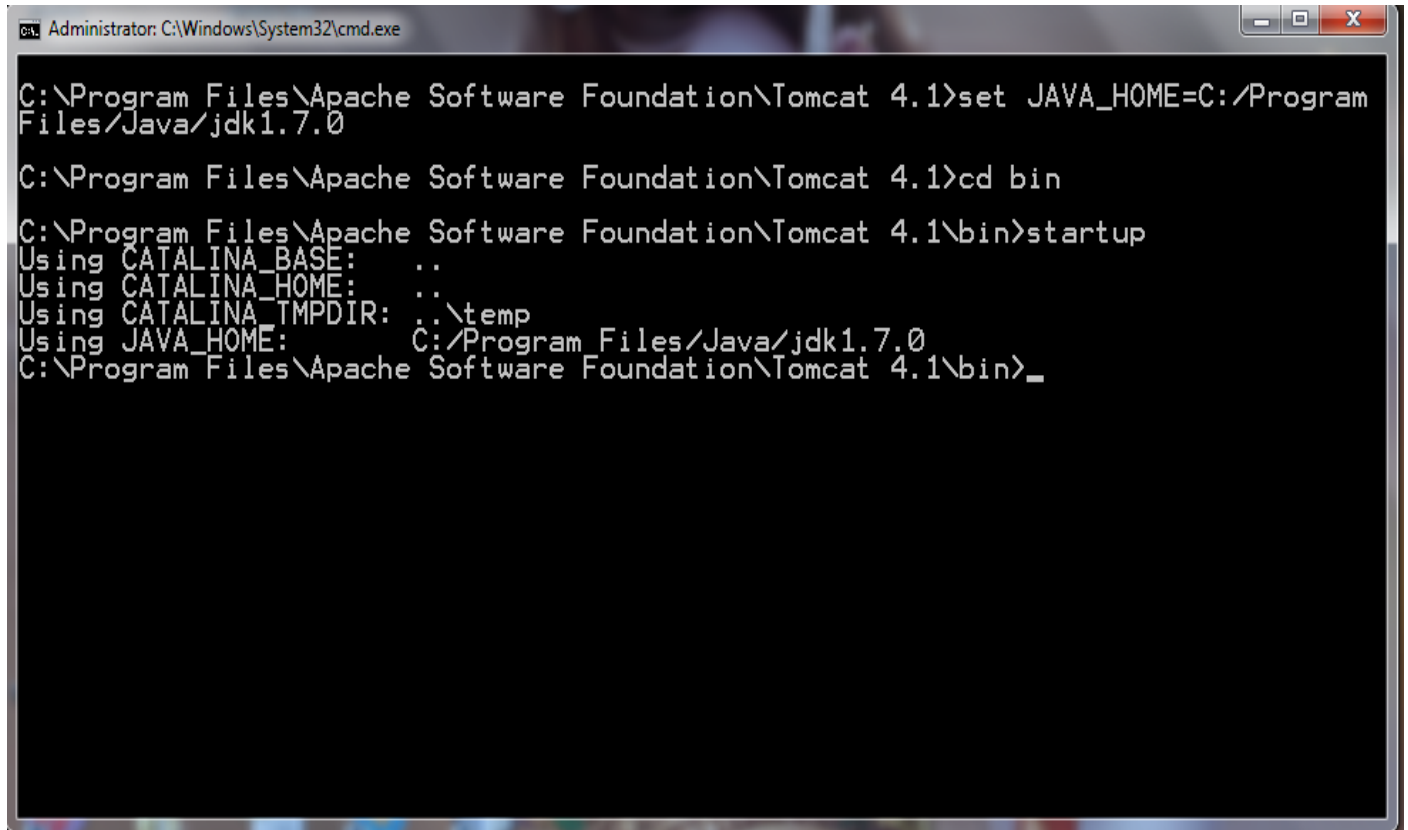
```
<html>
<head><title>cookies</title></head>
<body>
<form action="http://local host:8080/examples/servlet/cook" method="get">
FIRST NAME:<input type="text" name="fn"/></br>
LAST NAME:<input type="text" name="ln"/>
</br><input type="submit" value="SUBMIT"/>
</form></body></html>
```

Cook.java

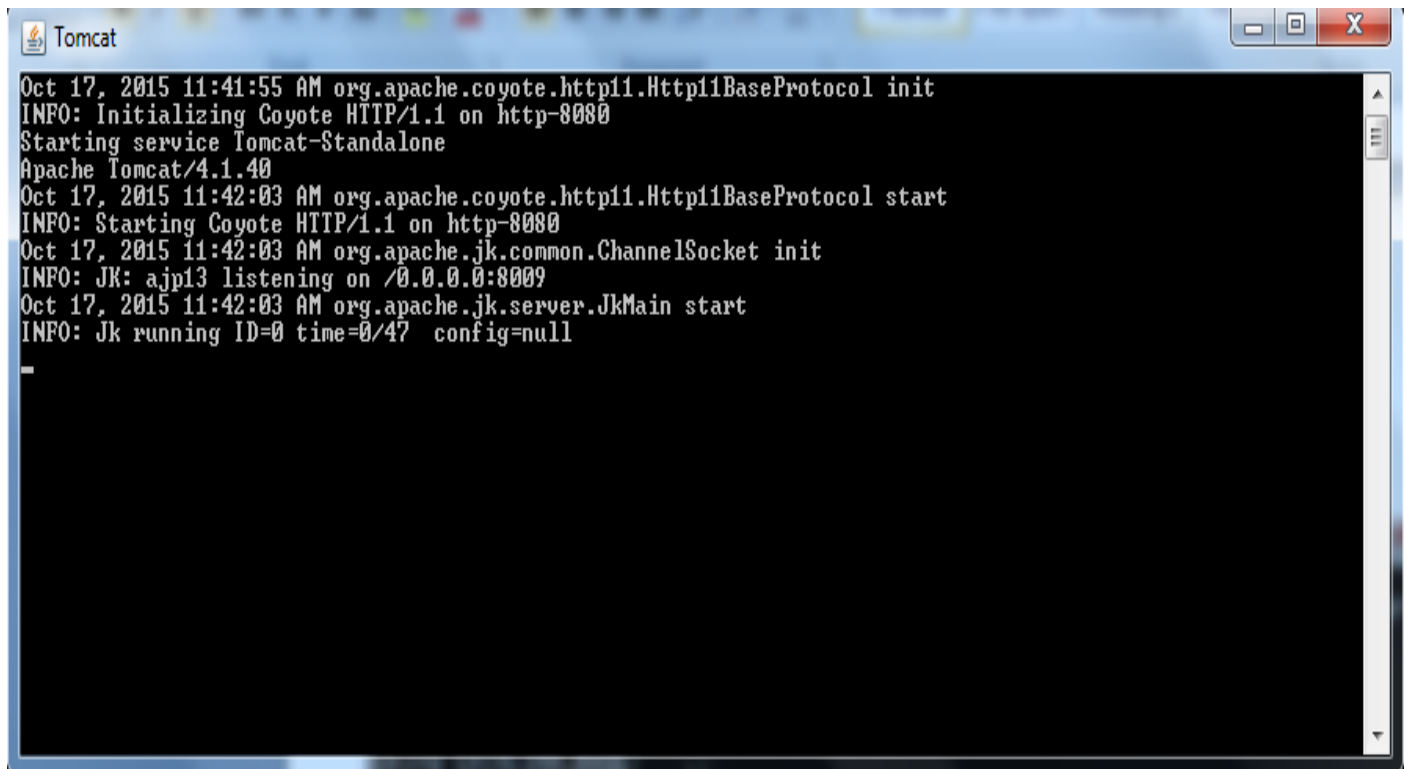
```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class cook extends HttpServlet
{
public void doGet(HttpServletRequest req,HttpServletResponse res)throws
ServletException,IOException
{
Cookie c1=new Cookie("cookie1",req.getParameter("fn"));
Cookie c2=new Cookie("cookie2",req.getParameter("ln"));
c1.setMaxAge(60*60*24);
c2.setMaxAge(60*60*24);
res.addCookie(c1);
res.addCookie(c2);
res.setContentType("text/html");
PrintWriter out=res.getWriter();
out.println("<center><font color='red'>SAMPLE COOKIES</font></center>");
out.println("<font color='green'>FIRST NAME:"+req.getParameter("fn")+</font>");
out.println("<font color='green'>LAST NAME:"+req.getParameter("ln")+</font>");
}
}
```


OUTPUT:

TOMCAT COMMAND WINDOW:



```
Administrator: C:\Windows\System32\cmd.exe
C:\Program Files\Apache Software Foundation\Tomcat 4.1>set JAVA_HOME=C:/Program
Files/Java/jdk1.7.0
C:\Program Files\Apache Software Foundation\Tomcat 4.1>cd bin
C:\Program Files\Apache Software Foundation\Tomcat 4.1\bin>startup
Using CATALINA_BASE:    ..
Using CATALINA_HOME:    ..
Using CATALINA_TMPDIR:  ..\temp
Using JAVA_HOME:        C:/Program Files/Java/jdk1.7.0
C:\Program Files\Apache Software Foundation\Tomcat 4.1\bin>_
```



```
Tomcat
Oct 17, 2015 11:41:55 AM org.apache.coyote.http11.Http11BaseProtocol init
INFO: Initializing Coyote HTTP/1.1 on http-8080
Starting service Tomcat-Standalone
Apache Tomcat/4.1.40
Oct 17, 2015 11:42:03 AM org.apache.coyote.http11.Http11BaseProtocol start
INFO: Starting Coyote HTTP/1.1 on http-8080
Oct 17, 2015 11:42:03 AM org.apache.jk.common.ChannelSocket init
INFO: JK: ajp13 listening on /0.0.0.0:8009
Oct 17, 2015 11:42:03 AM org.apache.jk.server.JkMain start
INFO: Jk running ID=0 time=0/47 config=null
-
```

JAVA COMMAND WINDOW:

```
C:\Program Files\Java\jdk1.7.0\bin>javac cook.java -classpath "C:\Program Files\nApache Software Foundation\Tomcat 4.1\common\lib\servlet.jar"
```

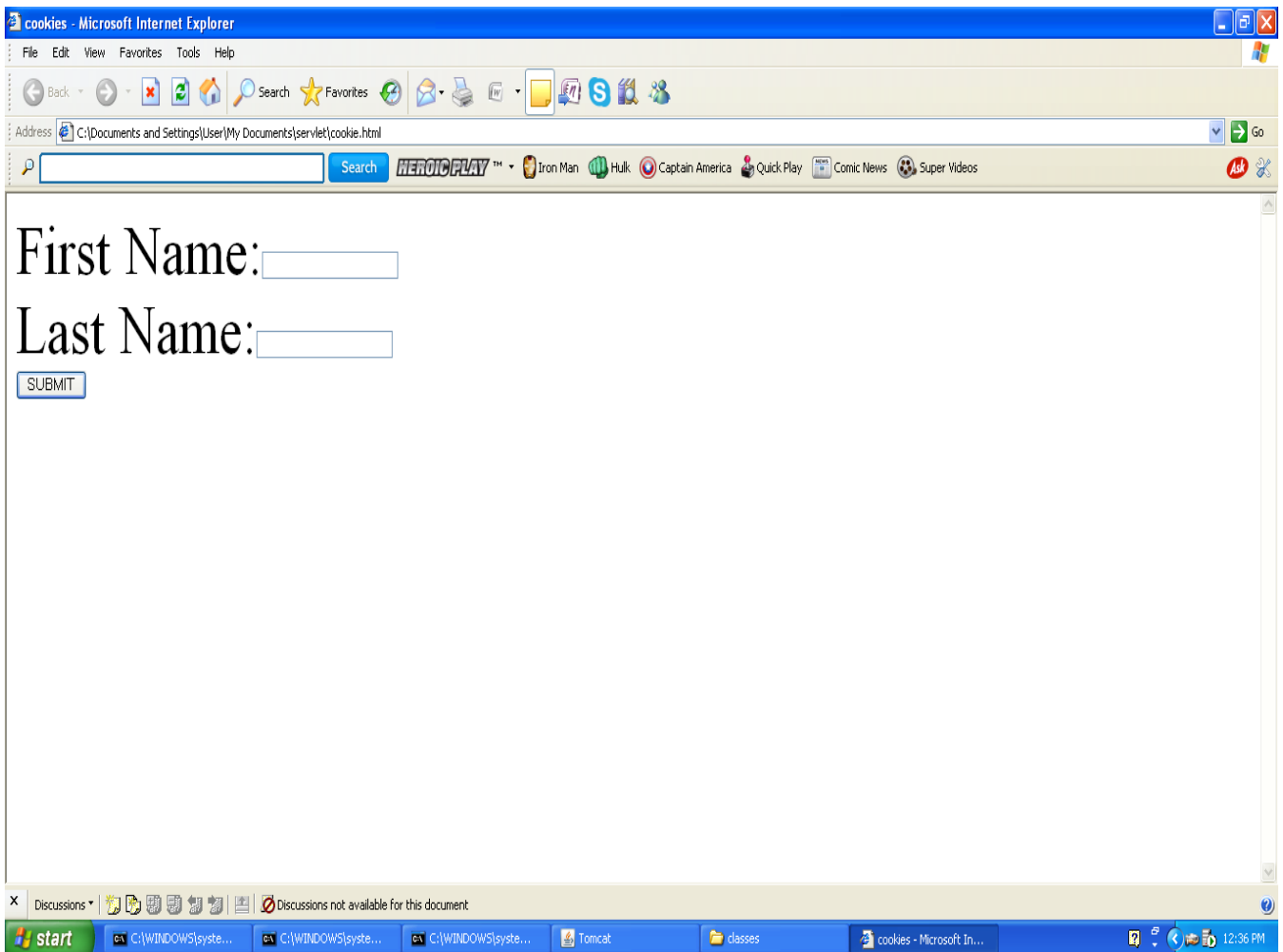
COPY THE FILE:

Local Disc(C:) \rightarrow Program Files \rightarrow Java \rightarrow jdk1.7.0 \rightarrow bin \rightarrow cook.class.

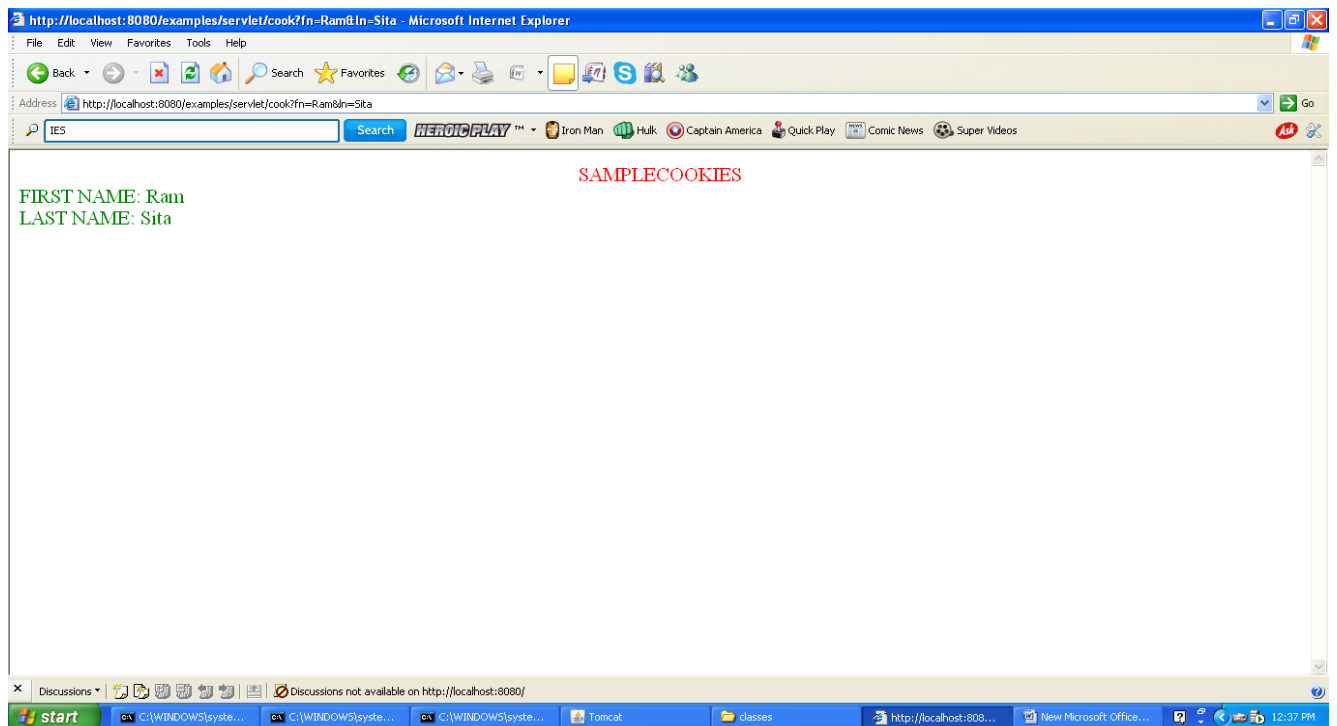
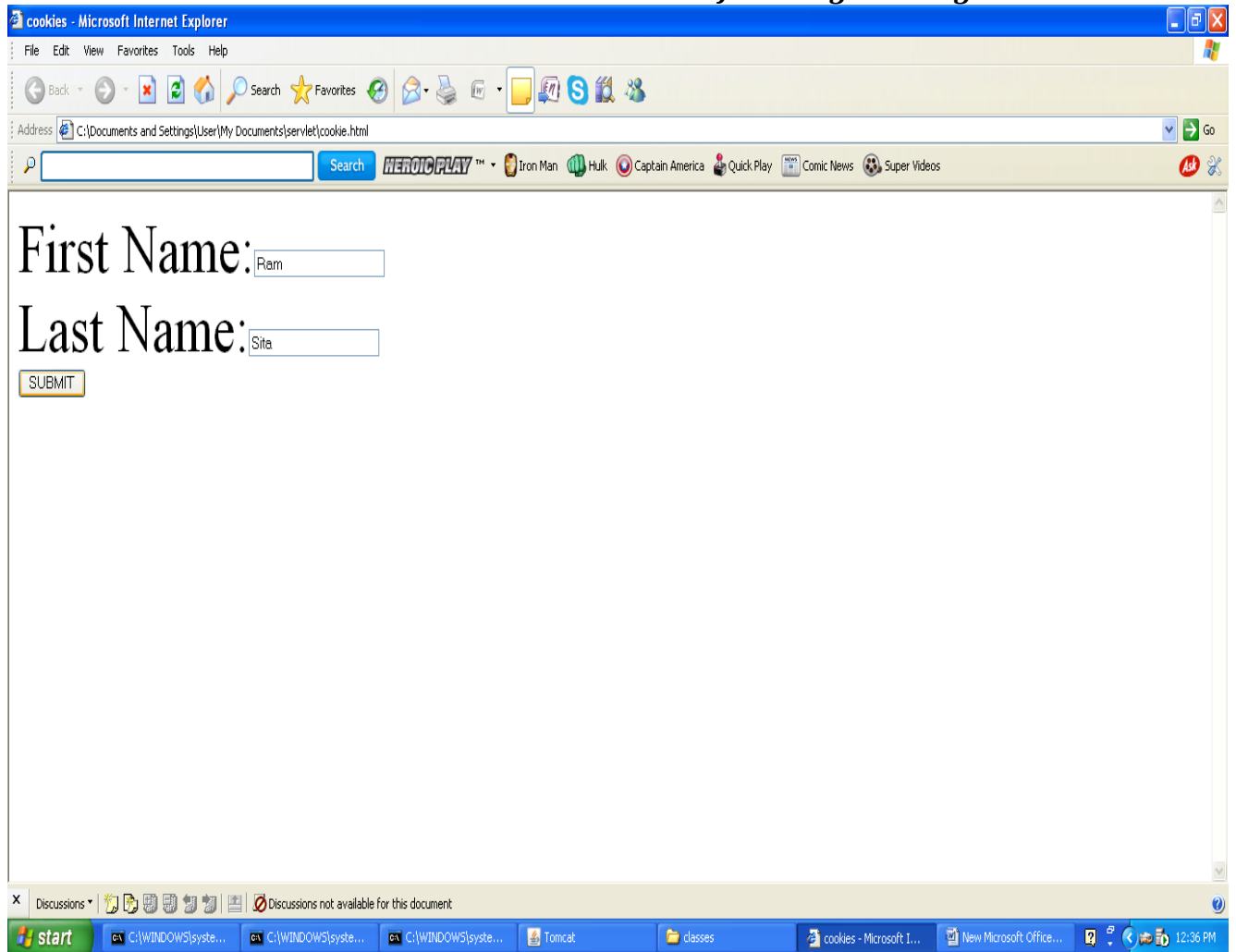
PASTE THE FILE:

Local Disc(C:) \rightarrow Program Files \rightarrow Apache Software Foundation \rightarrow Tomcat 4.1 \rightarrow Webapps \rightarrow Examples \rightarrow WEB_INF \rightarrow classes \rightarrow paste cook.class

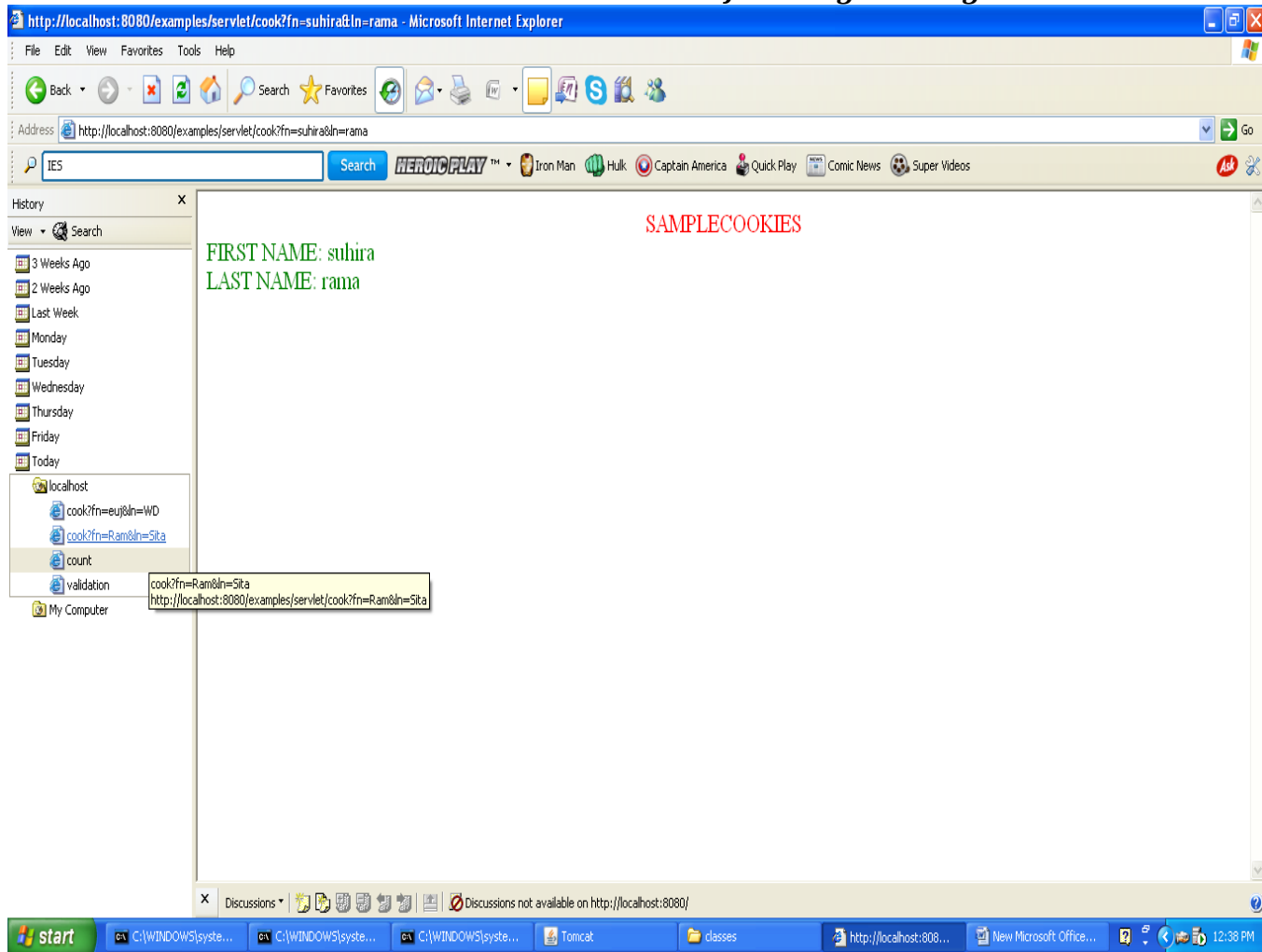
BROWSER WINDOW:



Advanced Java Programming Lab



Advanced Java Programming Lab



RESULT:

The above program has been executed successfully and the output was verified.

EX.NO:6

COUNT NUMBER OF ACCESS TIMES OF THE SERVLET PAGE

Aim:

To write java program to create Servlet to count the number of access time of that servlet page.

Algorithm:

Step 1:

Start the program.

Step 2:

Import the packages java.io. javax .servlet, javax.servlet.http .

Step 3:

Define class count extends from HttpServlet .

Step 4:

Set counting variable c as zero.

Step 5:

Define doGet () method. Set content type of page and initialize the object for PrintWriter class by calling the method getWriter ()

Step 6:

Increment the variable c by 1 for every access.

Step 7:

Print the value of c which indicates the counting.

Step 8:

Stop the program.

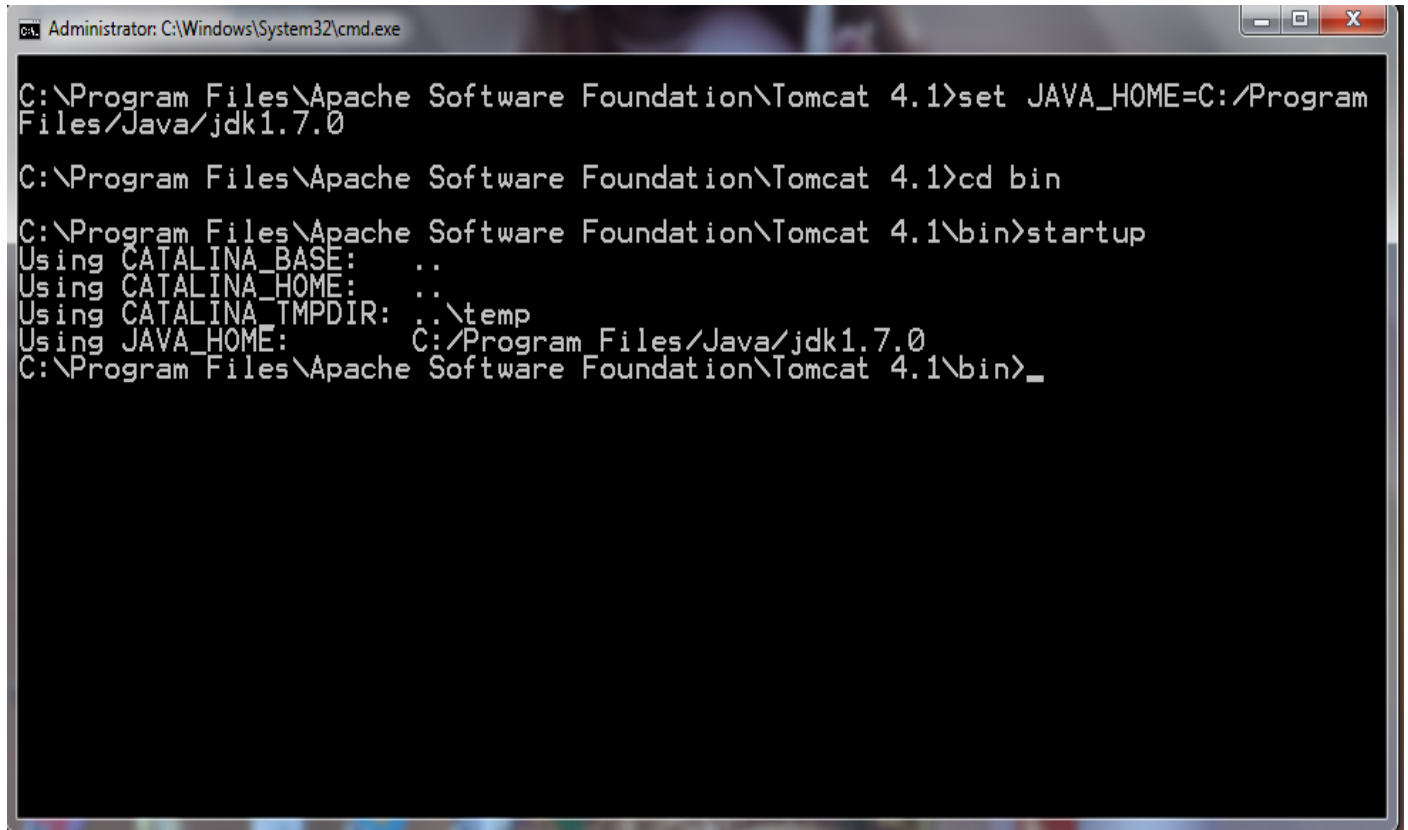
SOURCE CODE:

```
//Count Number of Access Times of the Servlet Page

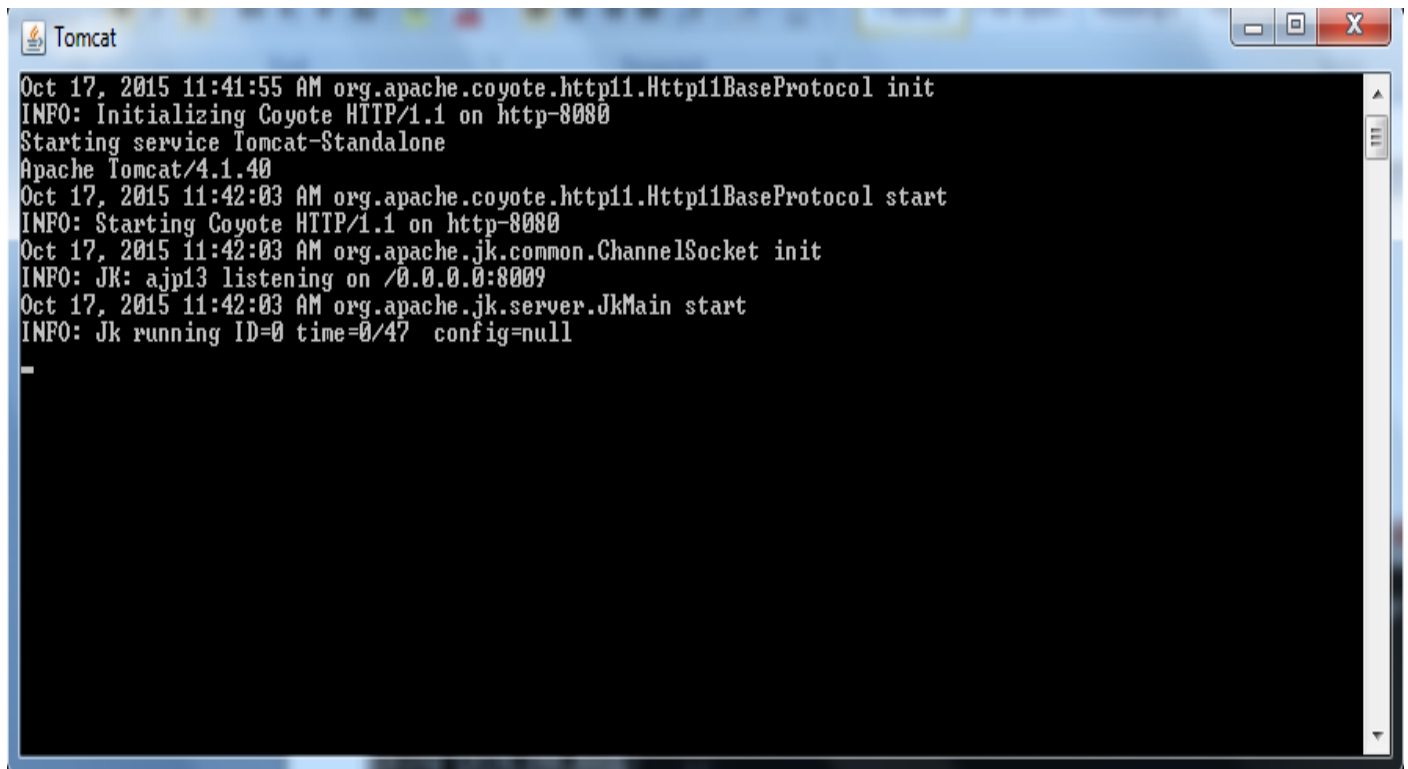
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class count extends HttpServlet
{
    int c=0;
    public void doGet(HttpServletRequest req,HttpServletResponse res)throws
        ServletException,IOException
    {
        res.setContentType("text/plain");
        PrintWriter out=res.getWriter();
        c++;
        out.println("since loading,this servlet hasbeen accessed"+c+"times");
    }
}
```

OUTPUT:

TOMCAT COMMAND WINDOW:



```
Administrator: C:\Windows\System32\cmd.exe
C:\Program Files\Apache Software Foundation\Tomcat 4.1>set JAVA_HOME=C:/Program
Files/Java/jdk1.7.0
C:\Program Files\Apache Software Foundation\Tomcat 4.1>cd bin
C:\Program Files\Apache Software Foundation\Tomcat 4.1\bin>startup
Using CATALINA_BASE:    ..
Using CATALINA_HOME:   ..
Using CATALINA_TMPDIR:  ..\temp
Using JAVA_HOME:       C:/Program Files/Java/jdk1.7.0
C:\Program Files\Apache Software Foundation\Tomcat 4.1\bin>_
```



```
Tomcat
Oct 17, 2015 11:41:55 AM org.apache.coyote.http11.Http11BaseProtocol init
INFO: Initializing Coyote HTTP/1.1 on http-8080
Starting service Tomcat-Standalone
Apache Tomcat/4.1.40
Oct 17, 2015 11:42:03 AM org.apache.coyote.http11.Http11BaseProtocol start
INFO: Starting Coyote HTTP/1.1 on http-8080
Oct 17, 2015 11:42:03 AM org.apache.jk.common.ChannelSocket init
INFO: JK: ajp13 listening on /0.0.0.0:8009
Oct 17, 2015 11:42:03 AM org.apache.jk.server.JkMain start
INFO: Jk running ID=0 time=0/47 config=null
-
```

JAVA COMMAND WINDOW:

```
C:\Program Files\Java\jdk1.7.0\bin>javac count.java -classpath "C:\Program Files\nApache Software Foundation\Tomcat 4.1\common\lib\servlet.jar"
```

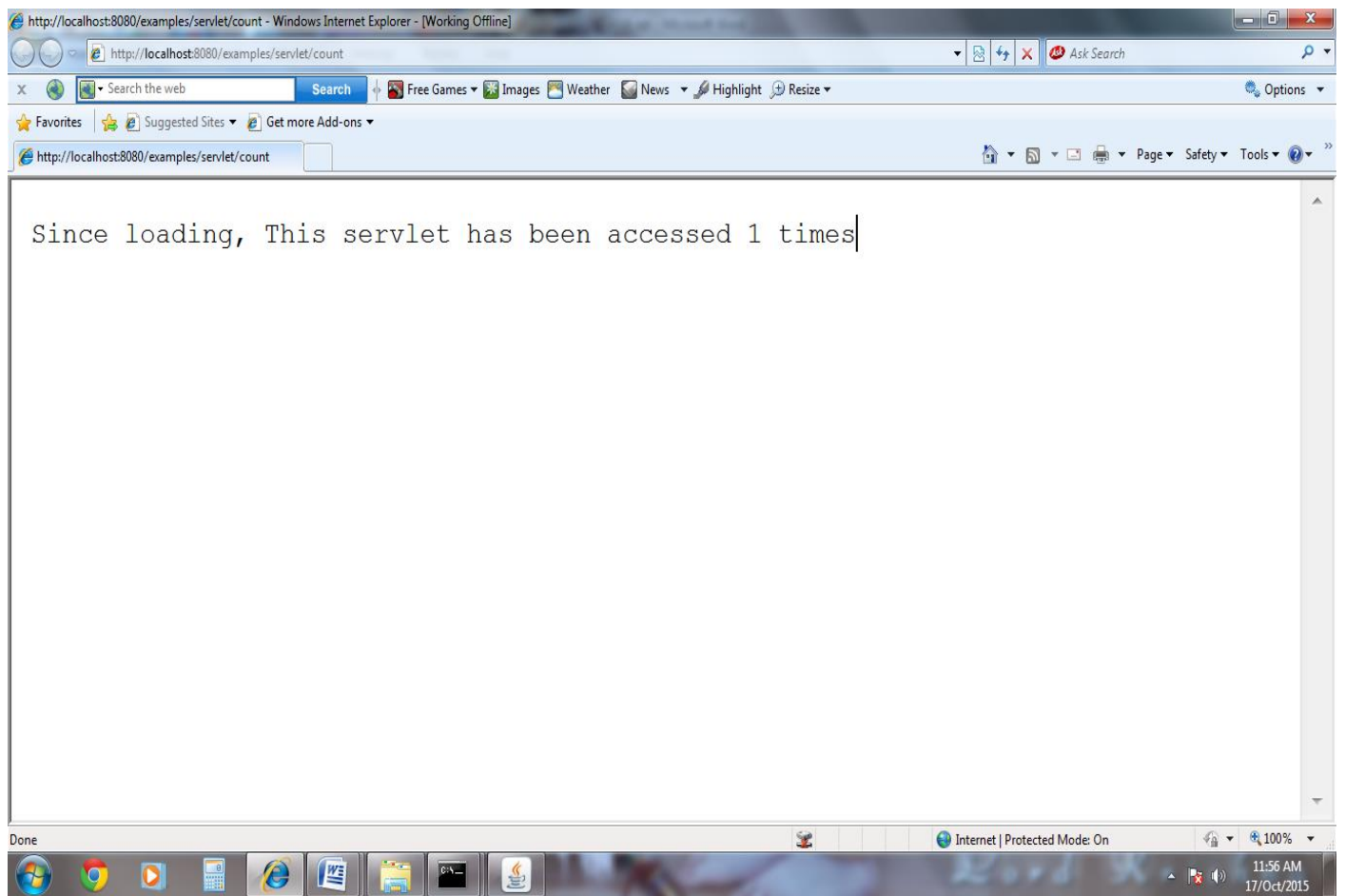
COPY THE FILE:

Local Disc(C:) \rightarrow Program Files \rightarrow Java \rightarrow jdk1.7.0 \rightarrow bin \rightarrow count.class.

PASTE THE FILE:

Local Disc(C:) \rightarrow Program Files \rightarrow Apache Software Foundation \rightarrow Tomcat 4.1 \rightarrow Webapps \rightarrow Examples \rightarrow WEB_INF \rightarrow classes \rightarrow paste count.class

BROWSER WINDOW:



RESULT:

The above program has been executed successfully and the output was verified.

EX.NO:7

CREATE A FORM AND VALIDATE PASSWORD USING SERVLET

Aim:

To write a java program to create a form and validate password using servlet.

Algorithm:

Step 1:

Start the program.

Step 2:

Create html file which contains the textbox for username and password with submit and reset button with alignment made by tables.

Step 3:

Create java program with class "validation"

Step 4:

Import required packages and extends the class from GenericServlet in main class.

Step 5:

Create object for printWriter () and read parameter () which is accessed by Httpservletrequest object.

Step 6:

Check username and password is "admin" if so, print the message "Welcome to this webpage", if not so, print the error message.

Step 7:

Close printWriter object.

Step 8:

Stop the program.

SOURCE CODE:

// Create a Form and Validate Password Using Servlet

Login.html:

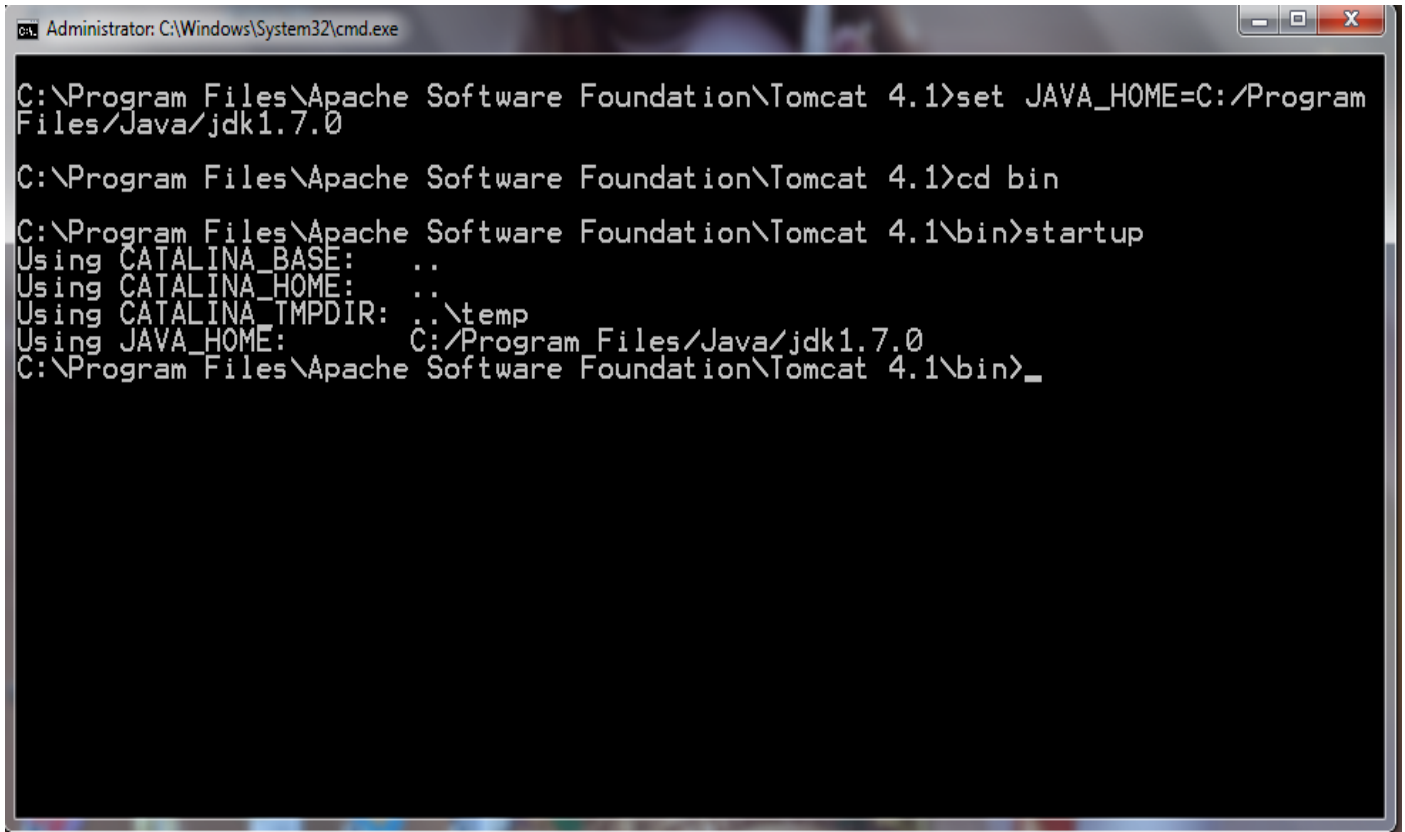
```
<html>
<head><title>login</title></head>
<body>
<form name="login form" method="post"
action="http://localhost:8080/examples/servlet/validation">
<br/><br/><br/><br/><br/>
<table align="center" border="3" border color="blue" cellspacing="0"height="120">
<tr><td align="center"><font color="blue" size="4">LOGIN FORM</font></td></tr>
<tr><td><table><tr><td>UserName</td><td><input type="text" name="user"/></td></tr>
<tr><td>Password</td><td><input type="password" name="pwd"/></td></tr>
<tr><td align="center"><input type="submit" value="LOGIN"/></td><td align="center"><input
type="Reset" value="RESET"/></td></tr>
</table></td></tr></table></form></body>
</html>
```

Validation.java:

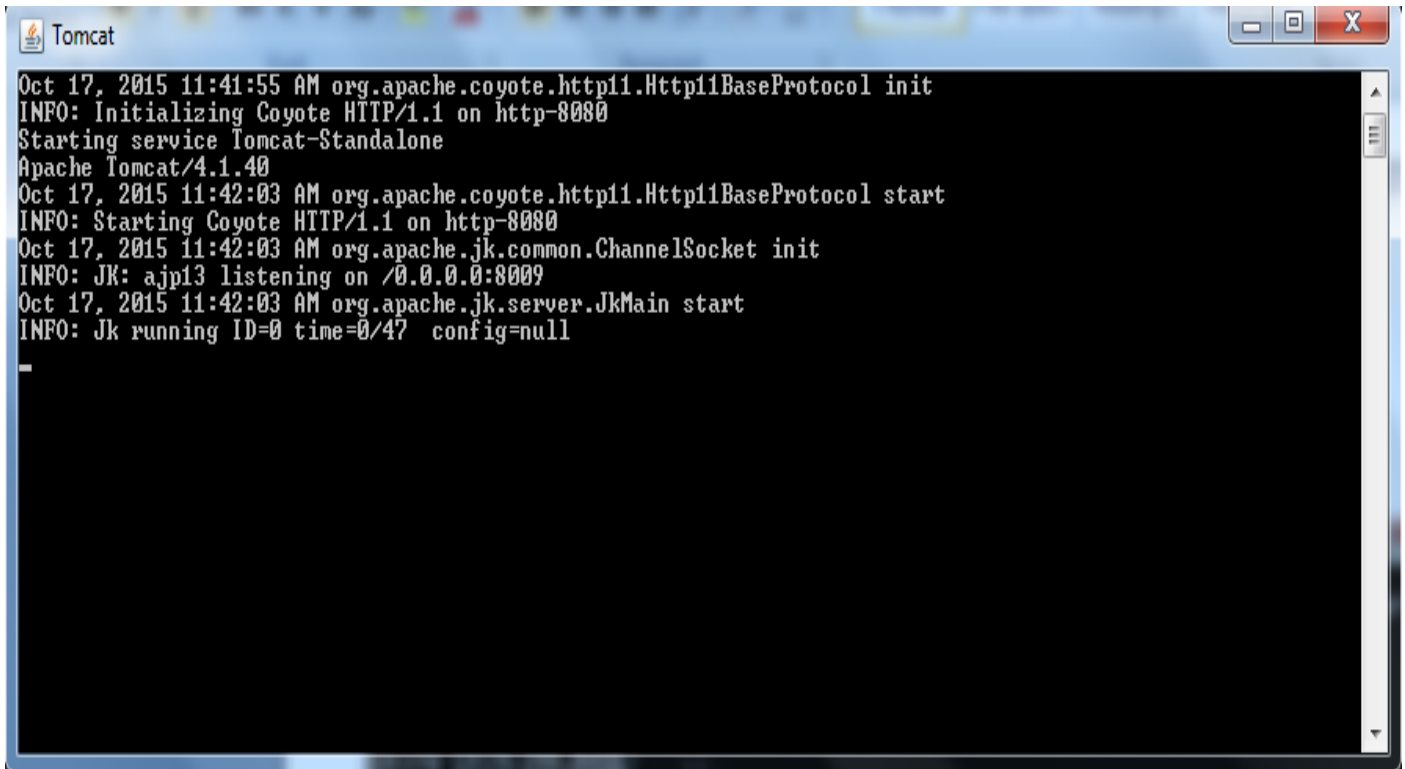
```
import java.io.*;
import java.util.*;
import javax.servlet.*;
public class validation extends GenericServlet
{
public void service(ServletRequest req,ServletResponse res)throws ServletException,IOException
{
PrintWriter pw=res.getWriter();
String x=req.getParameter("user");
String y=req.getParameter("pwd");
if(x.equals("admin")&&y.equals("admin"))
pw.println("<font color='green' size='5'>Welcome to this webpage</font>");
else
pw.println("<font color='red' size='5'>Invalid username or password</font>");
pw.close();
}
}
```

OUTPUT:

TOMCAT COMMAND WINDOW:



```
Administrator: C:\Windows\System32\cmd.exe
C:\Program Files\Apache Software Foundation\Tomcat 4.1>set JAVA_HOME=C:/Program
Files/Java/jdk1.7.0
C:\Program Files\Apache Software Foundation\Tomcat 4.1>cd bin
C:\Program Files\Apache Software Foundation\Tomcat 4.1\bin>startup
Using CATALINA_BASE:      ..
Using CATALINA_HOME:     ..
Using CATALINA_TMPDIR:   ..\temp
Using JAVA_HOME:         C:/Program Files/Java/jdk1.7.0
C:\Program Files\Apache Software Foundation\Tomcat 4.1\bin>_
```



```
Tomcat
Oct 17, 2015 11:41:55 AM org.apache.coyote.http11.Http11BaseProtocol init
INFO: Initializing Coyote HTTP/1.1 on http-8080
Starting service Tomcat-Standalone
Apache Tomcat/4.1.40
Oct 17, 2015 11:42:03 AM org.apache.coyote.http11.Http11BaseProtocol start
INFO: Starting Coyote HTTP/1.1 on http-8080
Oct 17, 2015 11:42:03 AM org.apache.jk.common.ChannelSocket init
INFO: JK: ajp13 listening on /0.0.0.0:8009
Oct 17, 2015 11:42:03 AM org.apache.jk.server.JkMain start
INFO: Jk running ID=0 time=0/47 config=null
-
```

JAVA COMMAND WINDOW:

C:\Program Files\Java\jdk1.7.0\bin>javac validation.java -classpath "C:\Program Files\Apache Software Foundation\Tomcat 4.1\common\lib\servlet.jar"

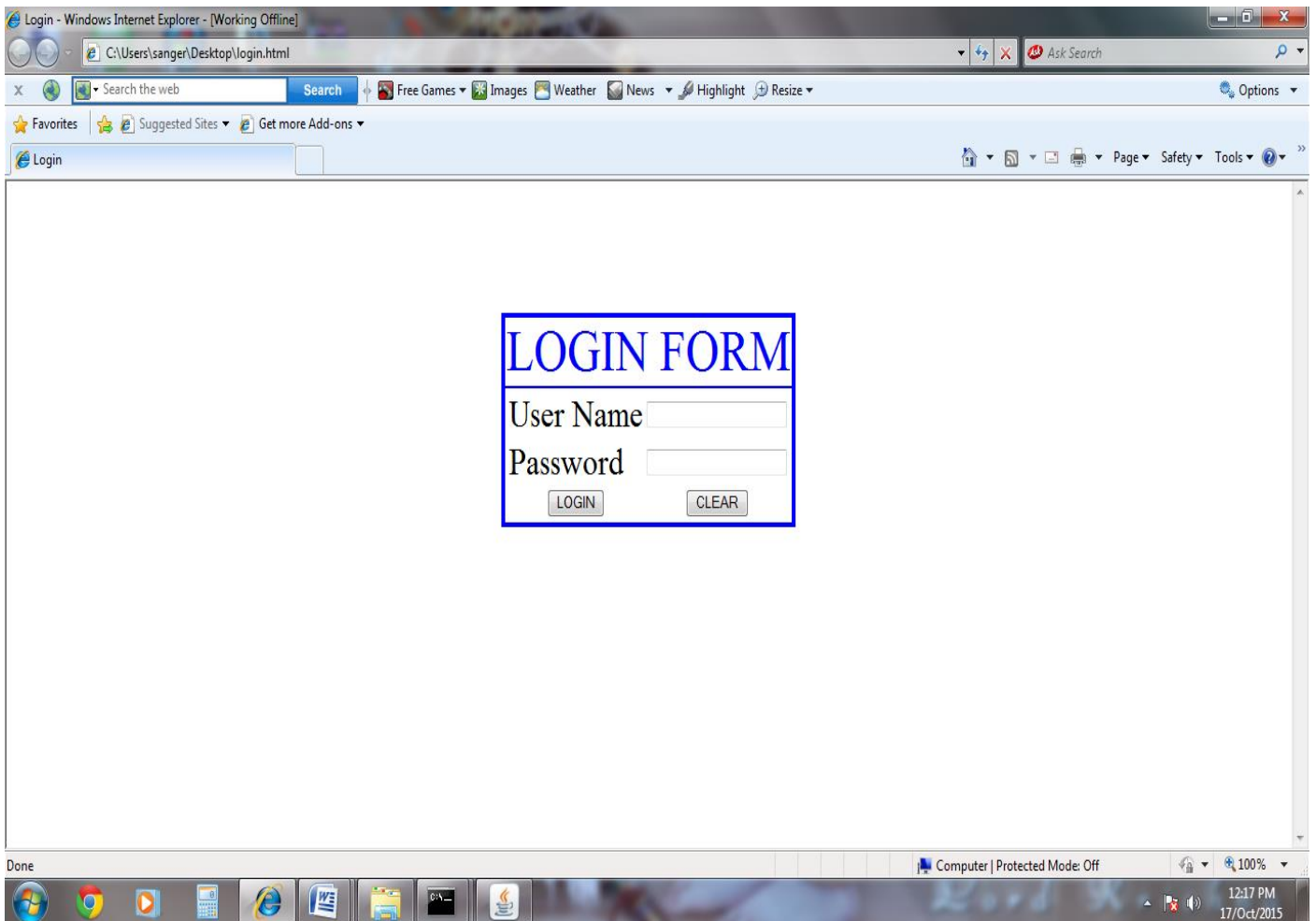
COPY THE FILE:

Local Disc(C:)->Program Files->Java->jdk1.7.0->bin->validation.class.

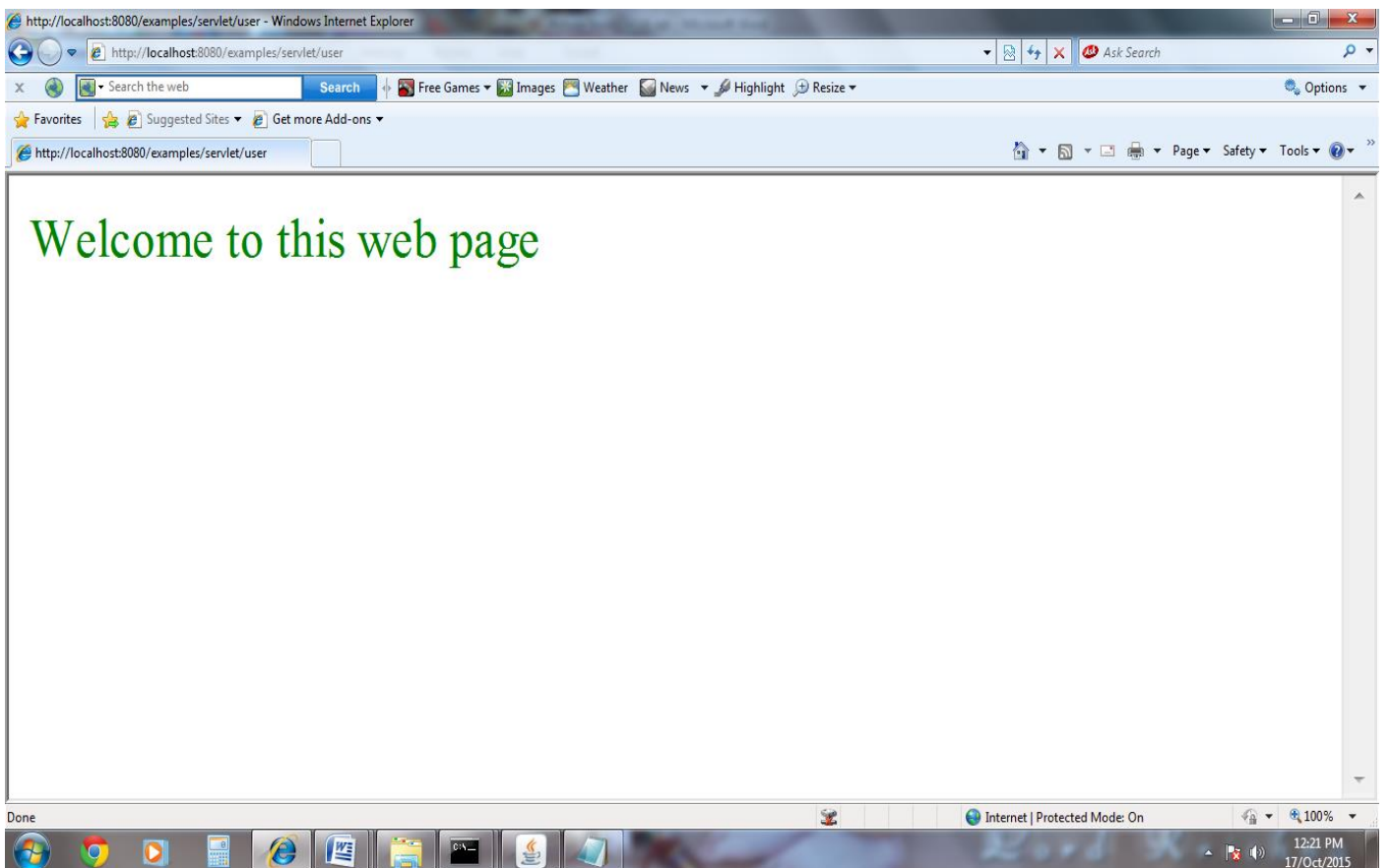
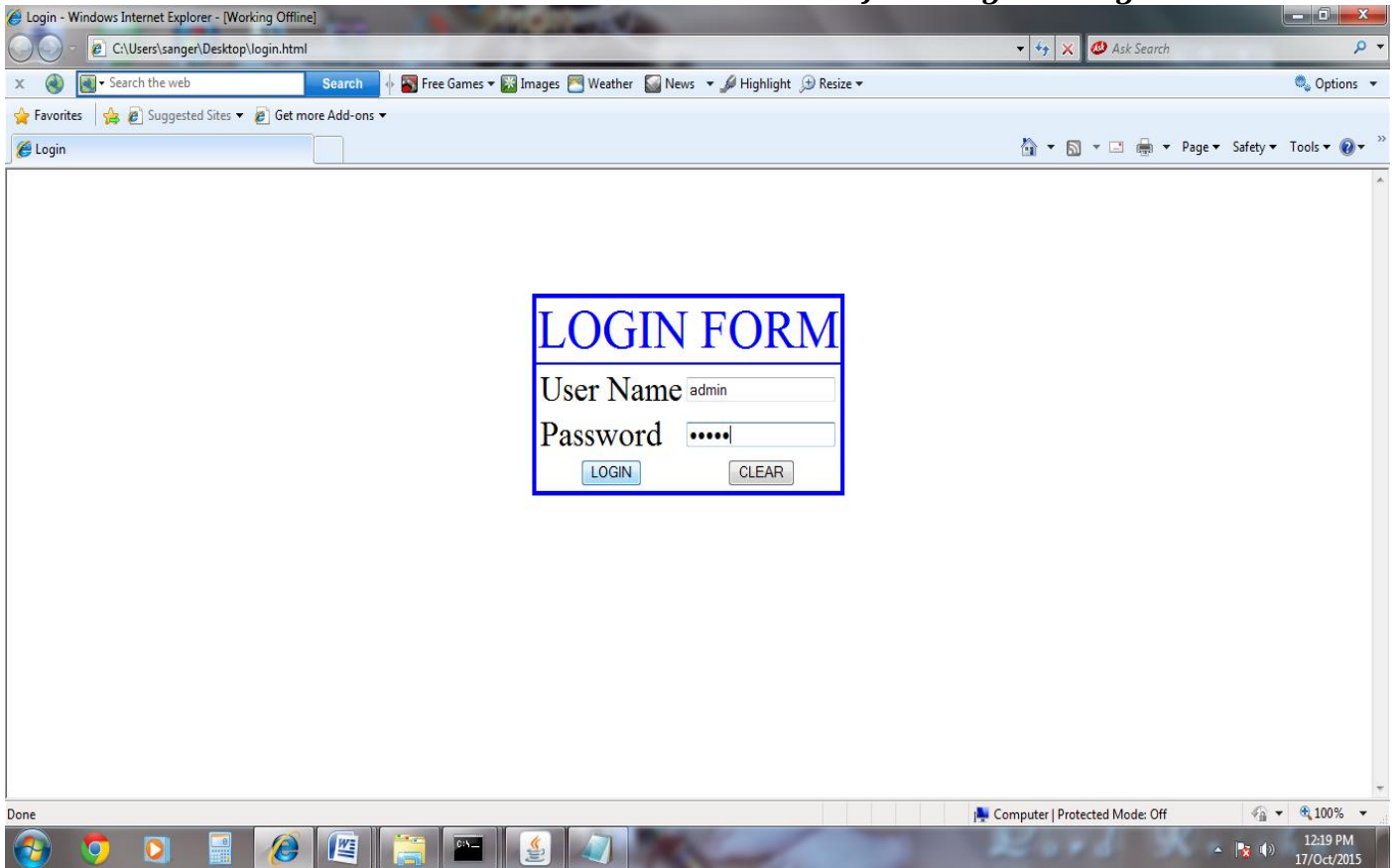
PASTE THE FILE:

Local Disc(C:) -> Program Files -> Apache Software Foundation -> Tomcat 4.1 -> Webapps -> Examples -> WEB_INF -> classes -> paste validation.class

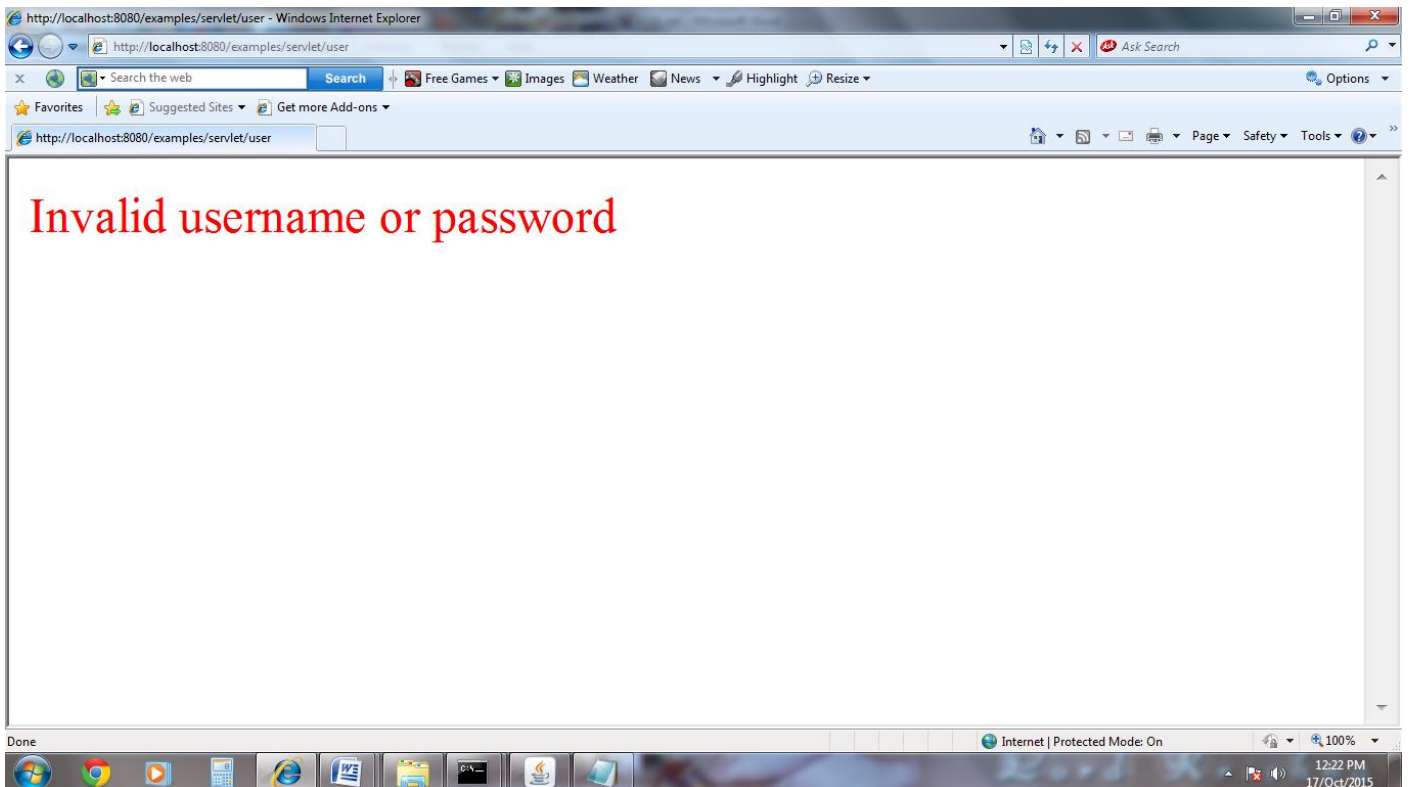
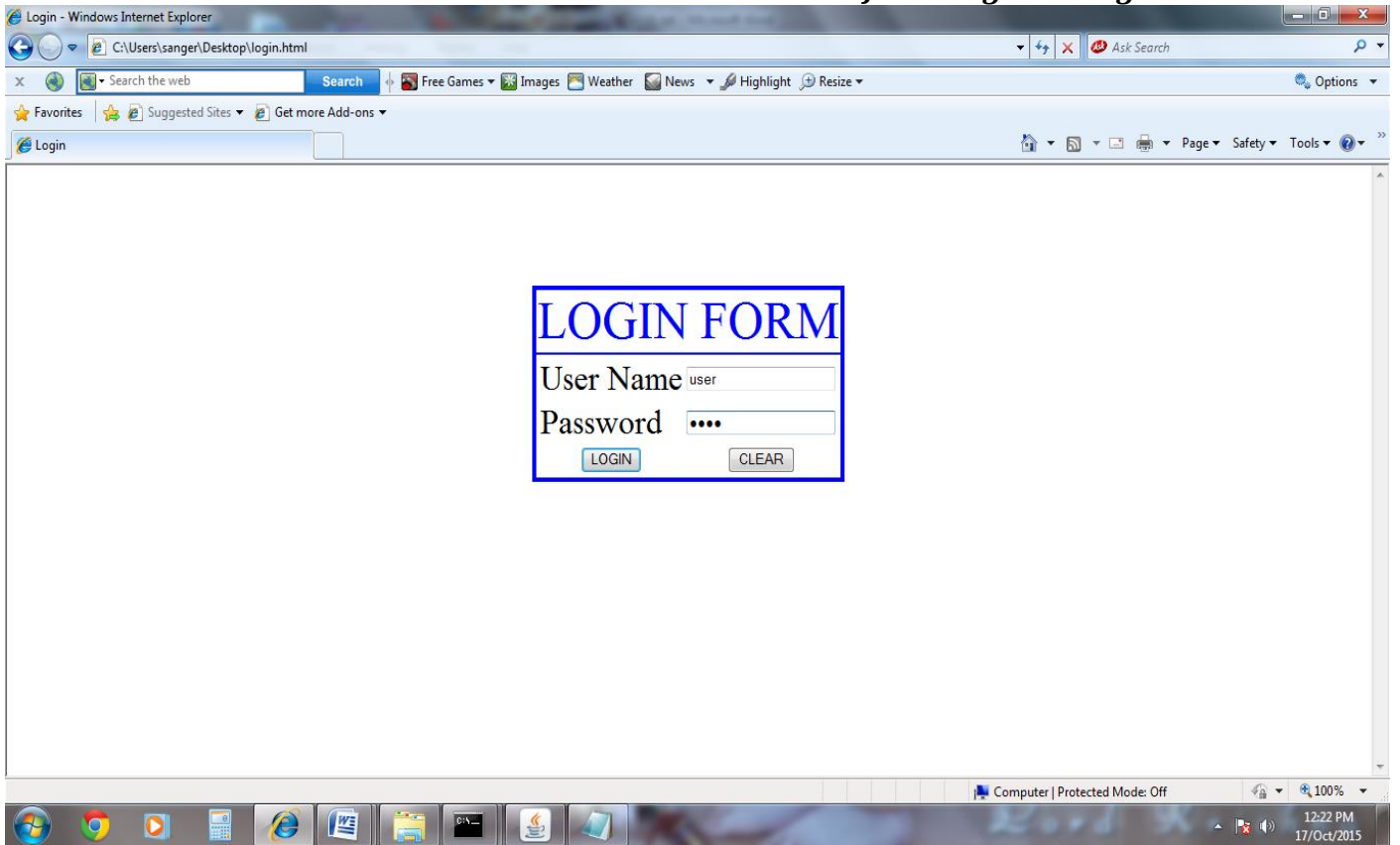
BROWSER WINDOW:



Advanced Java Programming Lab



Advanced Java Programming Lab



RESULT:

The above program has been executed successfully and the output was verified.

EX.NO:9

CONVERT AN IMAGE IN RGB TO A GRAYSCALE IMAGE

Aim:

Write a java program to convert an image in RGB to a grayscale image.

Algorithm:

Step 1:

Start the program.

Step 2:

Import the packages such as java.awt.*, java.awt.image.*, javax.imageio.ImageIO, java.io.*.

Step 3:

Define the main class rgb with main function and throws IOException.

Step 4:

Create object for DataInputStream, String class.

Step 5:

Create object for BufferedImage and read the input image file namely x1.jpg.

Step 6:

Take individual pixels of image and change the color of pixel by using the methods getRGB(), setRGB() with in for loop.

Step 7:

Get the output filename and stored the converted image within a filename by using ImageIO.write() function with the extension “.jpg”.

Step 8:

Stop the program.

SOURCE CODE:

```
//Convert an image in RGB to a Grayscale Image

import javax.imageio.ImageIO;
import java.awt.*;
import java.awt.image.*;
import java.io.*;

public class rgb
{
    public static void main(String args[]) throws IOException
    {
        DataInputStream in=new DataInputStream(System.in);
        String s;
        int w;
        BufferedImage img=ImageIO.read(new File("x1.jpg"));
        for(w=0;w<img.getWidth();w++)
        {
            for(int h=0;h<img.getHeight();h++)
            {
                Color oc=new Color(img.getRGB(w,h));
                int avg=((oc.getRed()+oc.getGreen()+oc.getBlue())/3);
                Color cc=new Color(avg,avg,avg);
                img.setRGB(w,h,cc.getRGB());
            }
        }
        System.out.println("Enter the Output FileName:");
        s=in.readLine();
        ImageIO.write(img,".jpg",new File(s+".jpg"));
        System.out.println("RGB image x1.jpg was successfully converted to Grayscale
                                                                    image stored in filename="+s+".jpg");
    }
}
```

OUTPUT:

```
C:\Program Files \Java\jdk1.7.0\bin>javac rgb.java
C:\Program Files \Java\jdk1.7.0\bin>java rgb
```

```
Enter the Output FileName:
x2
RGB image x1.jpg was successfully converted to Grayscale image stored in filename=x2.jpg
```

Notes: Go to Java Home Directory to verify the output.

RESULT:

The above program has been executed successfully and the output was verified.

EX.NO:10

DEVELOP CHAT SERVER USING JAVA

Aim:

To write java program for chat server using datagram packet and datagram socket.

Algorithm:

Step 1:

Start the program.

Step 2:

Import java.net package for client server application.

Step 3:

Define buffer size, server port, client port, byte array and declare object for datagram socket.

Step 4:

Create method theserver () for sending data to client using datagram packet object and send () method.

Step 5:

Create method theclient() for receiving packets from server using datagram packet object by receive() method.

Step 6:

Print data in client window by using getdata() and getlength() method.

Step 7:

Define main function and receive command line arguments.

Step 8:

If command line arguments length is 1 then call theserver() otherwise call theclient().

Step 9:

Stop the program.

SOURCE CODE:

```
// Develop Chat Server Using Java
```

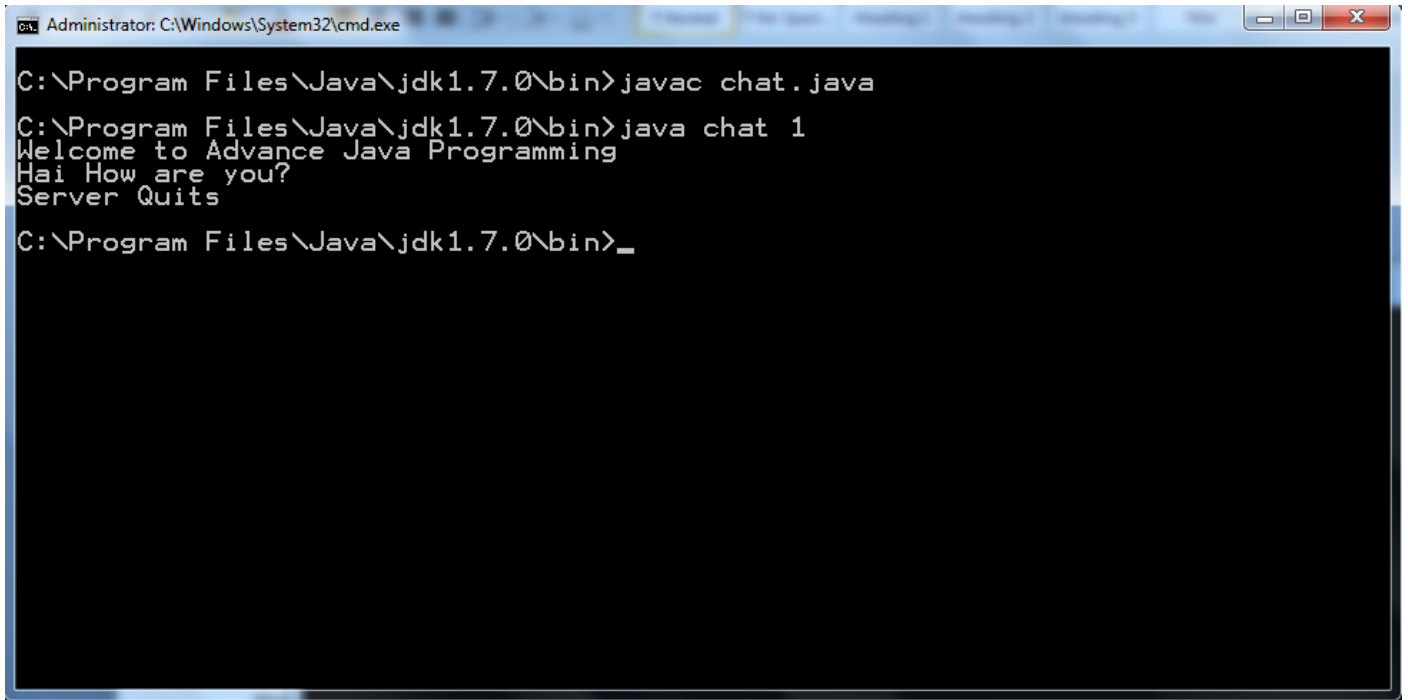
```
import java.net.*;
class chat
{
public static int BUFSIZE=1024;
public static int serverport=1057;
public static int clientport=1058;
public static DatagramSocket ds;
public static byte buffer[]=new byte[BUFSIZE];
public static void theserver()throws Exception
{
    int pos=0;
    while(true)
    {
    int c=System.in.read();
    switch(c)
    {
    case -1:
        System.out.println("Server quits");
        return;
    case '\n':
        DatagramPacket dp=new DatagramPacket (buffer,pos, InetAddress.getLocalHost(),
                                                clientport);

        ds.send(dp);
        pos=0;
        break;
    default:
        buffer[pos++]= (byte)c;
    }
    }
}
public static void theclient()throws Exception
{
    while(true)
    {
        DatagramPacket dp=new DatagramPacket(buffer,buffer.length);
        ds.receive(dp);
        String str=new String(dp.getData(),0,dp.getLength());
        System.out.println(str);
    }
}
public static void main(String ar[])throws Exception
{
    if(ar.length==1)
    {
        ds=new DatagramSocket(serverport);
        theserver();
    }
}
```

```
    }  
    else  
    {  
        ds=new DatagramSocket(clientport);  
        theclient();  
    }  
}  
}
```

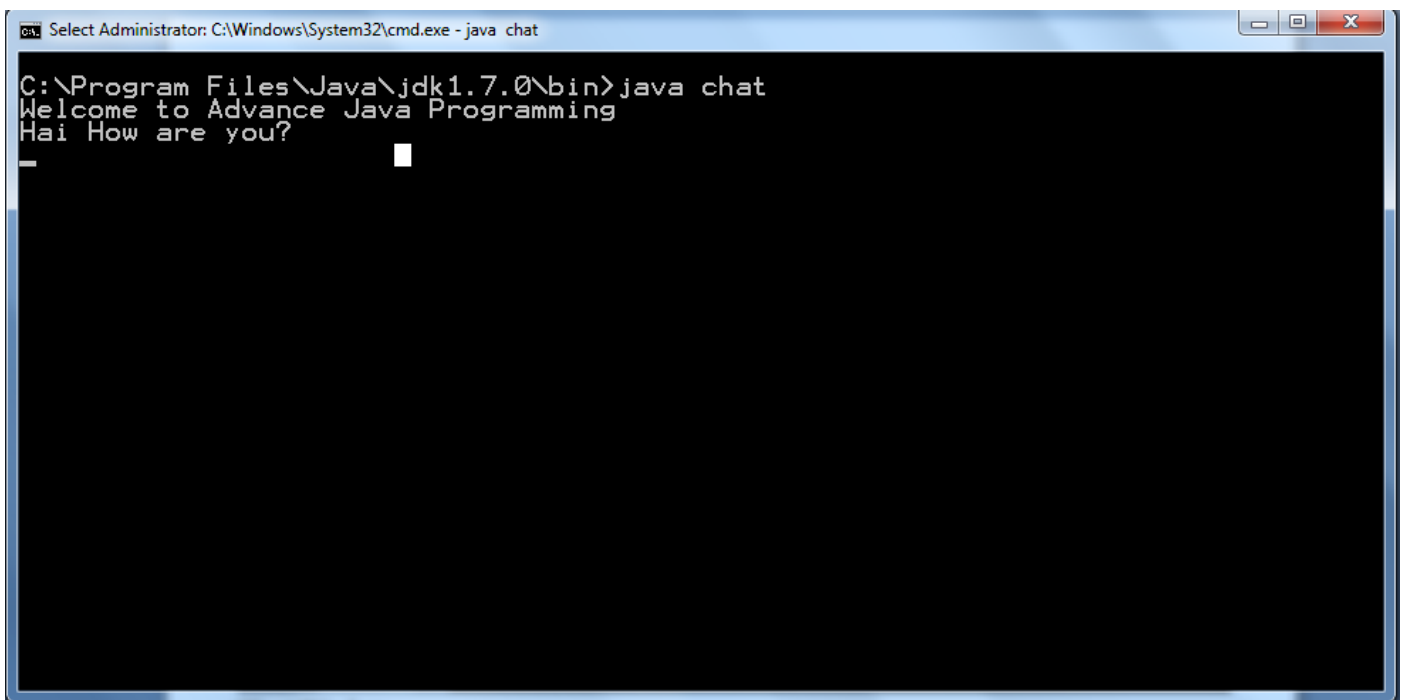
OUTPUT:

LOCALHOST SERVER SIDE COMMAND WINDOW



```
Administrator: C:\Windows\System32\cmd.exe
C:\Program Files\Java\jdk1.7.0\bin>javac chat.java
C:\Program Files\Java\jdk1.7.0\bin>java chat 1
Welcome to Advance Java Programming
Hai How are you?
Server Quits
C:\Program Files\Java\jdk1.7.0\bin>_
```

LOCALHOST CLIENT SIDE COMMAND WINDOW



```
Select Administrator: C:\Windows\System32\cmd.exe - java chat
C:\Program Files\Java\jdk1.7.0\bin>java chat
Welcome to Advance Java Programming
Hai How are you?
_
```

RESULT:

The above program has been executed successfully and the output was verified.