

Exercise 1

Implementation of Echo Server and Client Using TCP

EchoServer.java

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

public class EchoServer
{
    public static void main(String args[]) throws Exception
    {
        try
        {
            int Port;
            BufferedReader Buf =new BufferedReader(new InputStreamReader(System.in));
            System.out.print(" Enter the Port Address : " );
            Port=Integer.parseInt(Buf.readLine());
            ServerSocket sok =new ServerSocket(Port);
            System.out.println(" Server is Ready To Receive a Message. ");
            System.out.println(" Waiting ..... ");
            Socket so=sok.accept();
            if(so.isConnected()==true)
                System.out.println(" Client Socket is Connected Succcefully. ");
            InputStream in=so.getInputStream();
            OutputStream ou=so.getOutputStream();
            PrintWriter pr=new PrintWriter(ou);
            BufferedReader buf=new BufferedReader(new InputStreamReader(in));
            String str=buf.readLine();
            System.out.println(" Message Received From Client : " + str);
            System.out.print(" Enter message to Forwarded To Client: ");
            String msend=Buf.readLine();
            pr.println(msend);
            pr.flush();
        }
        catch(Exception e)
        {
            System.out.println(" Error : " + e.getMessage());
        }
    }
}
```

EchoClient.java

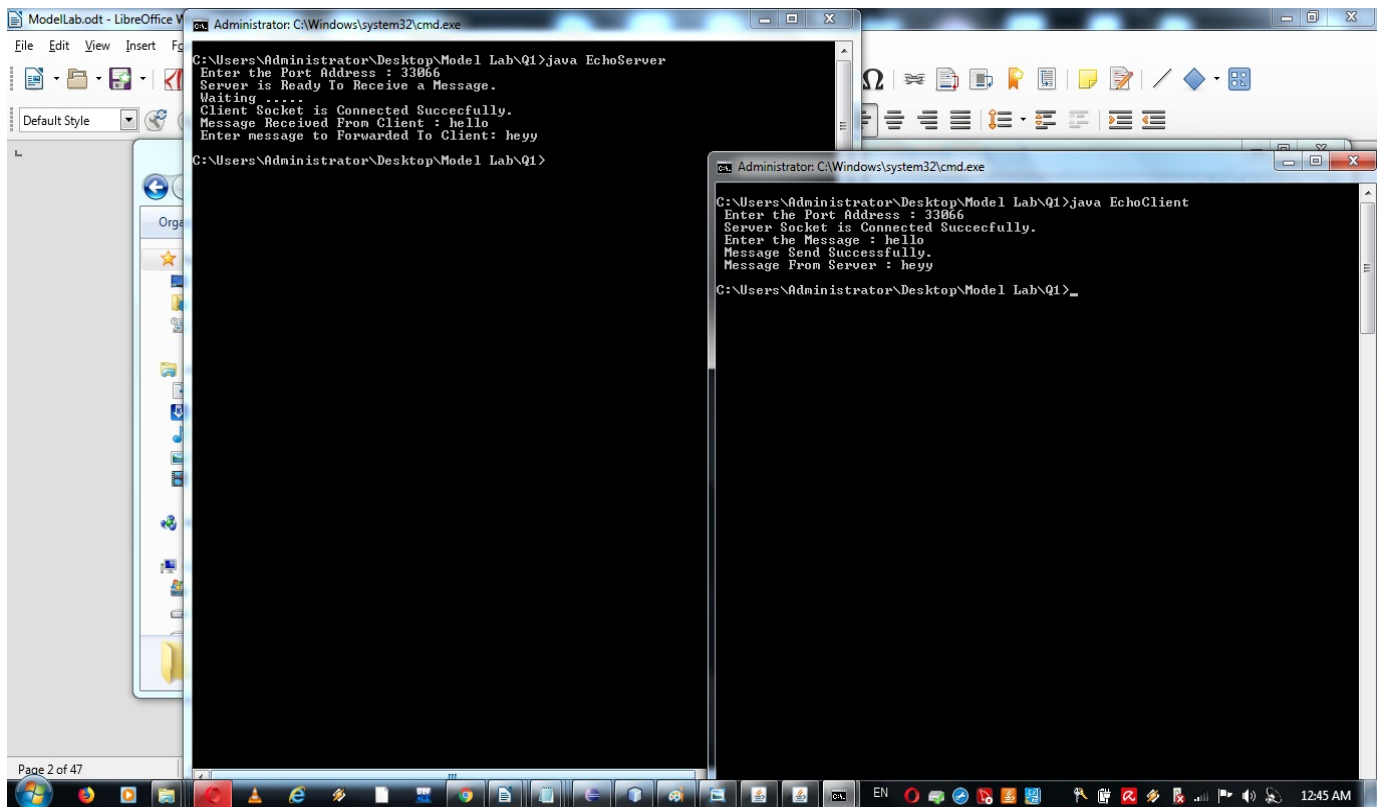
```
import java.io.*;
import java.net.*;
```

```

public class EchoClient
{
    public static void main(String args[]) throws Exception
    {
        try {
            int Port;
            BufferedReader Buf =new BufferedReader(new InputStreamReader(System.in));
            System.out.print(" Enter the Port Address : " );
            Port=Integer.parseInt(Buf.readLine());
            Socket sok=new Socket("localhost",Port);
            if(sok.isConnected()==true)
                System.out.println(" Server Socket is Connected Succcefully. ");
            InputStream in=sok.getInputStream();
            OutputStream ou=sok.getOutputStream();
            PrintWriter pr=new PrintWriter(ou);
            BufferedReader buf1=new BufferedReader(new InputStreamReader(System.in));
            BufferedReader buf2=new BufferedReader(new InputStreamReader(in));
            String str1,str2;
            System.out.print(" Enter the Message : ");
            str1=buf1.readLine();
            pr.println(str1);
            pr.flush();
            System.out.println(" Message Send Successfully. ");
            str2=buf2.readLine();
            System.out.println(" Message From Server : " + str2);
        }
        catch(Exception e)
        {
            System.out.println(" Error : " + e.getMessage());
        }
    }
}

```

Output:



Exercise 2

Echo Server with GUI in JAVA

ServerSide.java

```
import java.awt.BorderLayout;
import java.awt.EventQueue;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.JButton;
import javax.swing.JLabel;
import javax.swing.JTextField;
import java.awt.Color;
import java.awt.Font;
import javax.swing.JTextArea;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import java.io.*;
```

```

import java.net.*;

public class Serverside extends JFrame {

    private JPanel contentPane;
    private static JTextArea messageR;
    private static JLabel message_dis;
    private static Socket so;
    private static ServerSocket sok;
    private static DataInputStream in;
    private static DataOutputStream pr;

    private JTextArea message_field;

    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    Serverside frame = new Serverside();
                    frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });

        String str="Connect";
        try
    {
        int Port=33064;

        sok =new ServerSocket(Port);
        System.out.println(" Server is Ready To Receive a Message. ");
        System.out.println(" Waiting ..... ");
        so=sok.accept();
        if(so.isConnected()==true)
        {
            System.out.println(" Client Socket is Connected Succcefully. ");
            message_dis.setText("Client Socket is Connected Succcefully.");
        }
        in=new DataInputStream(so.getInputStream());
        pr=new DataOutputStream(so.getOutputStream());

        while(!str.equals("exit"))
        {

```

```

        str=in.readUTF();
        System.out.println(" Message Received From Client : " + str);
        messageR.setText(messageR.getText()+"\n"+str);
        pr.writeUTF("Message received: "+str);
    }

}

catch(Exception e)
{
    System.out.println("Server R");
    System.out.println(" Error : " + e.getMessage());
}

}

/**
 * Create the frame.
 */
public Serverside() {
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(0, 0, 907, 516);
    contentPane = new JPanel();
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
    setContentPane(contentPane);
    contentPane.setLayout(null);

    JButton btnSend = new JButton("Send");
    btnSend.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent arg0) {

            try
            {

                String msend="";
                msend=message_field.getText().toString();
                System.out.println(" Message send to dsffff Client : " + msend);
                pr.writeUTF(msend);

            }
            catch(Exception e)
            {
                System.out.println("Server S");
                System.out.println(e+" "+e.getMessage());
            }
        }
    });
}

```

```

        btnSend.setBounds(331, 218, 89, 23);
        contentPane.add(btnSend);

        JLabel lblEnterTheMessage = new JLabel("Enter the message");
        lblEnterTheMessage.setForeground(new Color(0, 102, 51));
        lblEnterTheMessage.setFont(new Font("Tahoma", Font.PLAIN, 14));
        lblEnterTheMessage.setBounds(85, 87, 171, 41);
        contentPane.add(lblEnterTheMessage);

        message_field = new JTextArea();
        message_field.setBounds(266, 97, 262, 78);
        contentPane.add(message_field);

        JLabel lblWelcomeToThe = new JLabel("Welcome to the Server");
        lblWelcomeToThe.setForeground(new Color(0, 102, 51));
        lblWelcomeToThe.setFont(new Font("Tahoma", Font.PLAIN, 14));
        lblWelcomeToThe.setBounds(341, 21, 171, 29);
        contentPane.add(lblWelcomeToThe);

        messageR = new JTextArea();
        messageR.setBounds(544, 97, 262, 78);
        contentPane.add(messageR);

        message_dis = new JLabel("");
        message_dis.setForeground(new Color(0, 102, 51));
        message_dis.setFont(new Font("Tahoma", Font.PLAIN, 14));
        message_dis.setBounds(378, 191, 262, 29);
        contentPane.add(message_dis);
    }
}

```

ClientSide.java

```

import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.EventQueue;
import java.awt.Font;

import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JTextArea;
import javax.swing.JTextField;
import javax.swing.border.EmptyBorder;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import java.io.*;
import java.net.*;

```

```

public class Clientside extends JFrame {

    private JPanel contentPane;
    private static JTextArea messageR;
    private static JLabel message_dis;
    private static Socket sok;
    private static DataInputStream in;
    private static DataOutputStream pr;

    private JTextArea message_field;

    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    Clientside frame = new Clientside();
                    frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });

        String str="Connected";
        try {

            int Port=33064;

            sok=new Socket("localhost",Port);
            if(sok.isConnected()==true)
                System.out.println(" Server Socket is Connected Succcefully. ");

            in=new DataInputStream(sok.getInputStream());
            pr=new DataOutputStream(sok.getOutputStream());

            while(!str.equals("exit"))
            {
                str=in.readUTF();
                System.out.println(str);
                messageR.setText(messageR.getText()+"\n"+str);
            }

        }
    }
}

```

```

catch(Exception e)
{
    System.out.println("Clinet R");
    System.out.println(" Error : " + e.getMessage());
}

}

/**
 * Create the frame.
 */
public Clientside() {
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(300, 300, 894, 464);
    contentPane = new JPanel();
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
    setContentPane(contentPane);
    contentPane.setLayout(null);

    JButton send_btn = new JButton("Send");
    send_btn.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent arg0) {

            try
            {

                String str1="";
                // System.out.print(" Enter the Message : ");
                // message_dis.setText("Enter the message");
                str1=message_field.getText().trim();
                System.out.println("jjffkgkfd"+ str1);
                pr.writeUTF(str1);

            }
            catch(Exception e)
            {
                System.out.println("Clinet S");
                System.out.println(e+" "+e.getMessage());
            }
        }
    });
    send_btn.setBounds(333, 196, 89, 23);
    contentPane.add(send_btn);

    JLabel lblEnterTheMessage = new JLabel("Enter the message");
    lblEnterTheMessage.setForeground(new Color(0, 102, 51));
    lblEnterTheMessage.setFont(new Font("Tahoma", Font.PLAIN, 14));
    lblEnterTheMessage.setBounds(69, 97, 171, 41);

```



```
contentPane.add(lblEnterTheMessage);
```

```
message_field = new JTextArea();  
message_field.setBounds(254, 107, 262, 78);  
contentPane.add(message_field);
```

```
JLabel lblWelcomeToThe = new JLabel("Welcome to the Client");  
lblWelcomeToThe.setForeground(new Color(0, 102, 51));  
lblWelcomeToThe.setFont(new Font("Tahoma", Font.PLAIN, 14));  
lblWelcomeToThe.setBounds(343, 11, 171, 29);  
contentPane.add(lblWelcomeToThe);
```

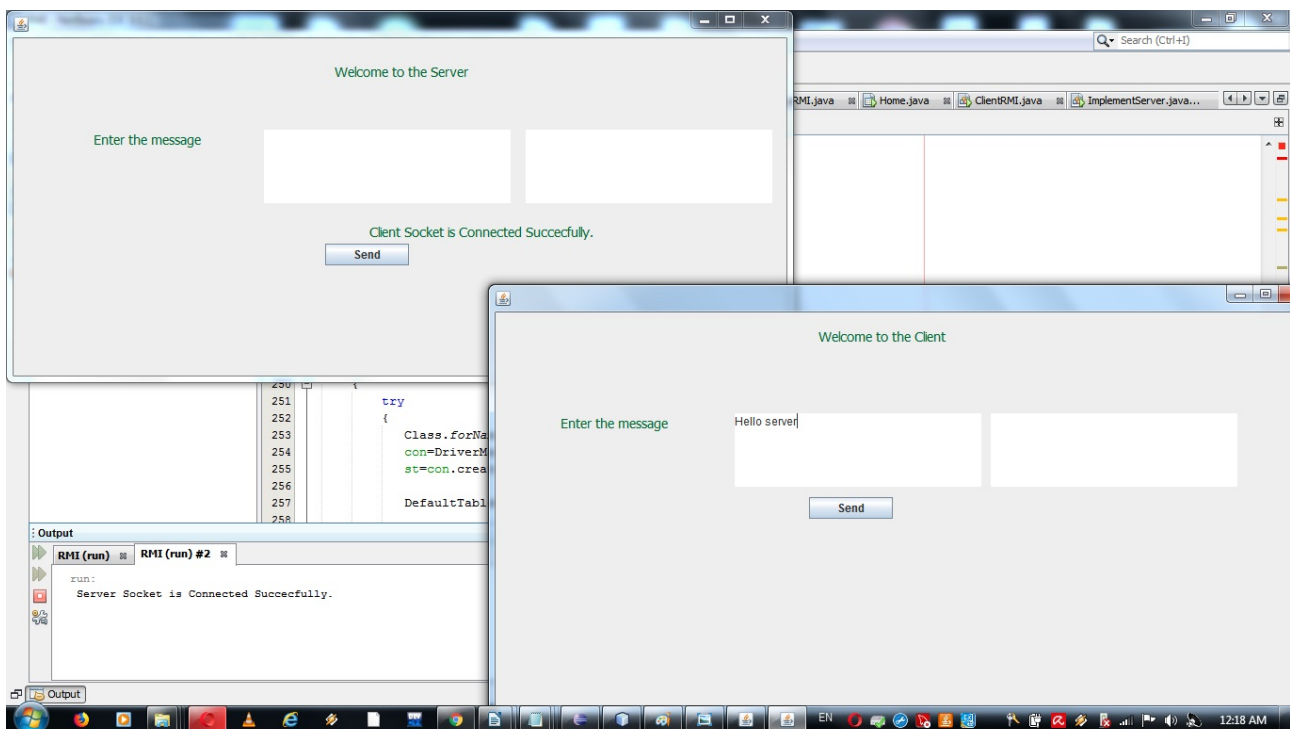
```
messageR = new JTextArea();  
messageR.setBounds(526, 107, 262, 78);  
contentPane.add(messageR);
```

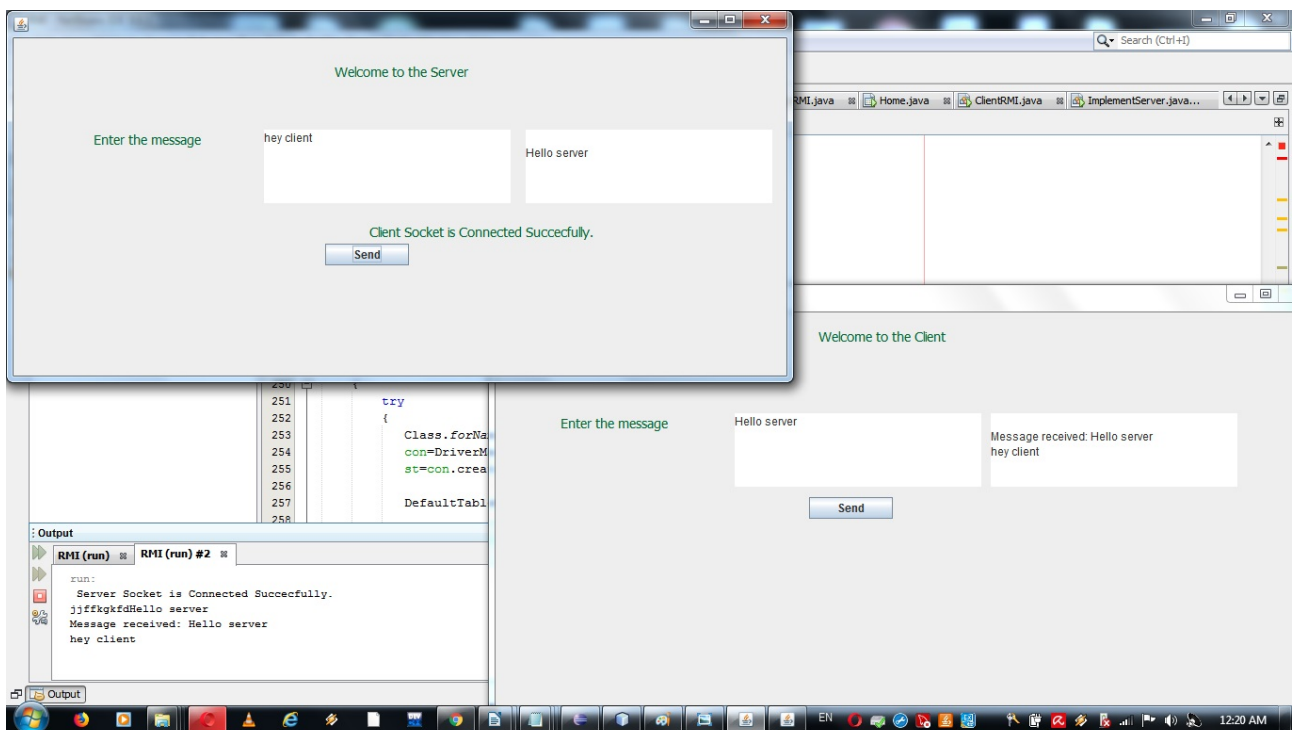
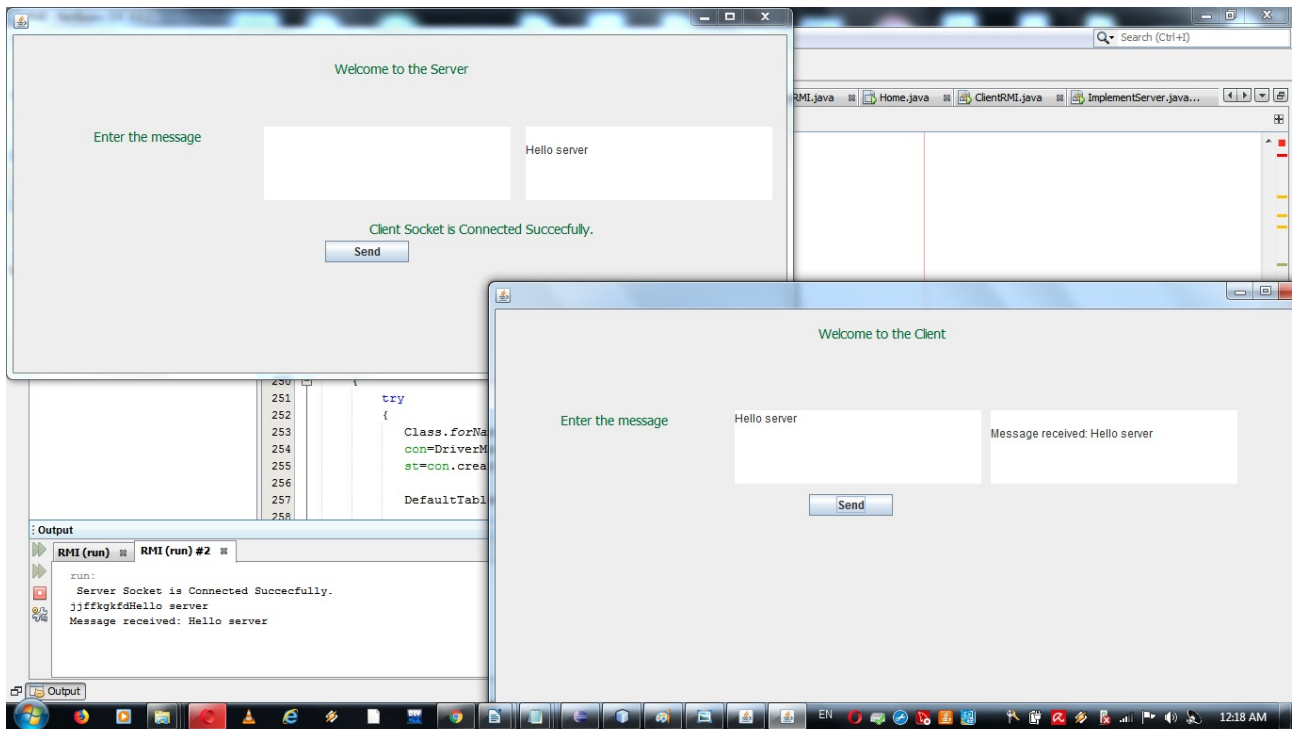
```
message_dis = new JLabel("");  
message_dis.setForeground(new Color(0, 102, 51));  
message_dis.setFont(new Font("Tahoma", Font.PLAIN, 14));  
message_dis.setBounds(374, 190, 262, 29);  
contentPane.add(message_dis);
```

```
}
```

```
}
```

Output:





Exercise 3

Write both client and server programs demonstrating Socket API using multithreaded client/server in java.

Server.java

```
import java.io.IOException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.util.Scanner;

public class Server
{
    public static void main(String[] args) throws IOException
    {
        DatagramSocket ds = new DatagramSocket(33066);
        System.out.println("Server");

        while (true)
        {
            DatagramPacket DpReceive = null;
            byte[] receive = new byte[65535];
            DpReceive = new DatagramPacket(receive, receive.length);
            ds.receive(DpReceive);
            ClientHandler obj = new ClientHandler(ds,DpReceive,receive);
            obj.start();
        }
    }
}

class ClientHandler extends Thread
{
    final DatagramSocket ds;
    DatagramPacket DpReceive;
    byte []receive;
    public ClientHandler( DatagramSocket ds,DatagramPacket DpReceive,byte []receive)
    {
        this.ds = ds;
        this.DpReceive=DpReceive;
        this.receive=receive;
    }
}
```

```

public void run()
{
    System.out.println("A new client is connected : ");
    try
    {
        InetAddress ip = DpReceive.getAddress();
        int clientport=DpReceive.getPort();
        byte buf[] = null;
        System.out.println("Client: " + data(receive));
        String inp = Calculate1(receive);

        buf = inp.getBytes();
        DatagramPacket DpSend = new DatagramPacket(buf, buf.length, ip, clientport);

        ds.send(DpSend);
        receive = new byte[65535];
    }
    catch (Exception e)
    {
        System.out.println(e.getLocalizedMessage());
    }
}

```

```

public static String Calculate1(byte[] a)
{
    if (a == null)
        return null;
    int i = 0;
    int num=0;
    String inval="Invalid number";
    int num1=0;
    int num2=0;
    int s1=1;
    int s2=1;
    String nums1="";
    String nums2="";
    char op;
    if(a[i]=='-')
    {
        s1*=-1;
        i++;
    }
    while(a[i]!=0 &&(a[i]>='0'&&a[i]<='9'))
    {
        nums1+=(char)a[i];
        i++;
    }
    if(nums1.length()==0)
        return inval;
}

```

```

if(a[i]=='-'||a[i]=='+'||a[i]=='*'||a[i]=='/')
{
    op=(char)a[i];
    i++;
}
else
    return inval;
if(a[i]=='-')
{
    s2*=-1;
    i++;
}

while(a[i]!=0&&(a[i]>='0'&&a[i]<='9'))
{
    nums2+=(char)a[i];
    i++;
}
if(nums2.length()==0)
    return inval;
if(a[i]!=0)
    return inval;
//System.out.println(nums1);
//System.out.println(nums2);
num1=Integer.parseInt(nums1);
num2=Integer.parseInt(nums2);
num1=num1*s1;
num2=num2*s2;
switch(op)
{
    case '+':
        num=num1+num2;
        break;
    case '-':
        num=num1-num2;
        break;
    case '*':
        num=num1*num2;
        break;
    case '/':
        {
            if(num2==0)
                return "Division by 0 is not possible";
            num=num1/num2;
        }
}
}

```

```

        System.out.println(num);
        String res=String.valueOf(num);
        return res;
    }
    public static StringBuilder data(byte[] a)
    {
        if (a == null)
            return null;
        StringBuilder ret = new StringBuilder();
        int i = 0;
        while (a[i] != 0)
        {
            ret.append((char) a[i]);
            i++;
        }
        return ret;
    }
}

```

ClientUDP.java

```

// Surendra Lodhi
import java.io.IOException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.util.Scanner;

public class ClientUDP
{
    public static void main(String args[]) throws IOException
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("CLient");

        DatagramSocket ds = new DatagramSocket();

        InetAddress ip = InetAddress.getLocalHost();
        byte buf[] = null;

        byte[] receive = new byte[65535];
        DatagramPacket DpReceive = null;
        while (true)
        {
            System.out.println("Enter the number:");
            String inp = sc.nextLine();

```

```

    buf = inp.getBytes();

    DatagramPacket DpSend =
        new DatagramPacket(buf, buf.length, ip, 33066);

    ds.send(DpSend);

    DpReceive = new DatagramPacket(receive, receive.length);

    ds.receive(DpReceive);

    System.out.println("From Server to Client: " + data(receive));

    if (inp.equals("stop"))
        break;
    receive = new byte[65535];

}
}
public static StringBuilder data(byte[] a)
{
    if (a == null)
        return null;
    StringBuilder ret = new StringBuilder();
    int i = 0;
    while (a[i] != 0)
    {
        ret.append((char) a[i]);
        i++;
    }
    return ret;
}
}

```

Client1.java

```

import java.io.IOException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.util.Scanner;

```

```

public class Client1
{
    public static void main(String args[]) throws IOException
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("CLient1:");

        DatagramSocket ds = new DatagramSocket();

        InetAddress ip = InetAddress.getLocalHost();
        byte buf[] = null;

        byte[] receive = new byte[65535];
        DatagramPacket DpReceive = null;
        while (true)
        {
            System.out.println("Enter the number:");
            String inp = sc.nextLine();

            buf = inp.getBytes();

            DatagramPacket DpSend = new DatagramPacket(buf, buf.length, ip, 33066);
            ds.send(DpSend);

            DpReceive = new DatagramPacket(receive, receive.length);

            ds.receive(DpReceive);
            System.out.println("From Server to Client: " + data(receive));

            if (inp.equals("stop"))
                break;

            receive = new byte[65535];
        }
    }
    public static StringBuilder data(byte[] a)
    {
        if (a == null)
            return null;
        StringBuilder ret = new StringBuilder();
        int i = 0;
        while (a[i] != 0)
        {
            ret.append((char) a[i]);
        }
    }
}

```



```

        i++;
    }
    return ret;
}
}

```

Client2.java

```

import java.io.IOException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.util.Scanner;

public class Client2
{
    public static void main(String args[]) throws IOException
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("CLient");
        // Step 1:Create the socket object for
        // carrying the data.
        DatagramSocket ds = new DatagramSocket();

        InetAddress ip = InetAddress.getLocalHost();
        byte buf[] = null;

        // loop while user not enters "bye"
        byte[] receive = new byte[65535];
        System.out.println("Server");
        DatagramPacket DpReceive = null;
        while (true)
        {
            System.out.println("Enter the message:");
            String inp = sc.nextLine();

            // convert the String input into the byte array.
            buf = inp.getBytes();

            // Step 2 : Create the datagramPacket for sending
            // the data.
            DatagramPacket DpSend =
                new DatagramPacket(buf, buf.length, ip, 33066);

            // Step 3 : invoke the send call to actually send
            // the data.
            ds.send(DpSend);
        }
    }
}

```

```

DpReceive = new DatagramPacket(receive, receive.length);

// Step 3 : review the data in byte buffer.
ds.receive(DpReceive);

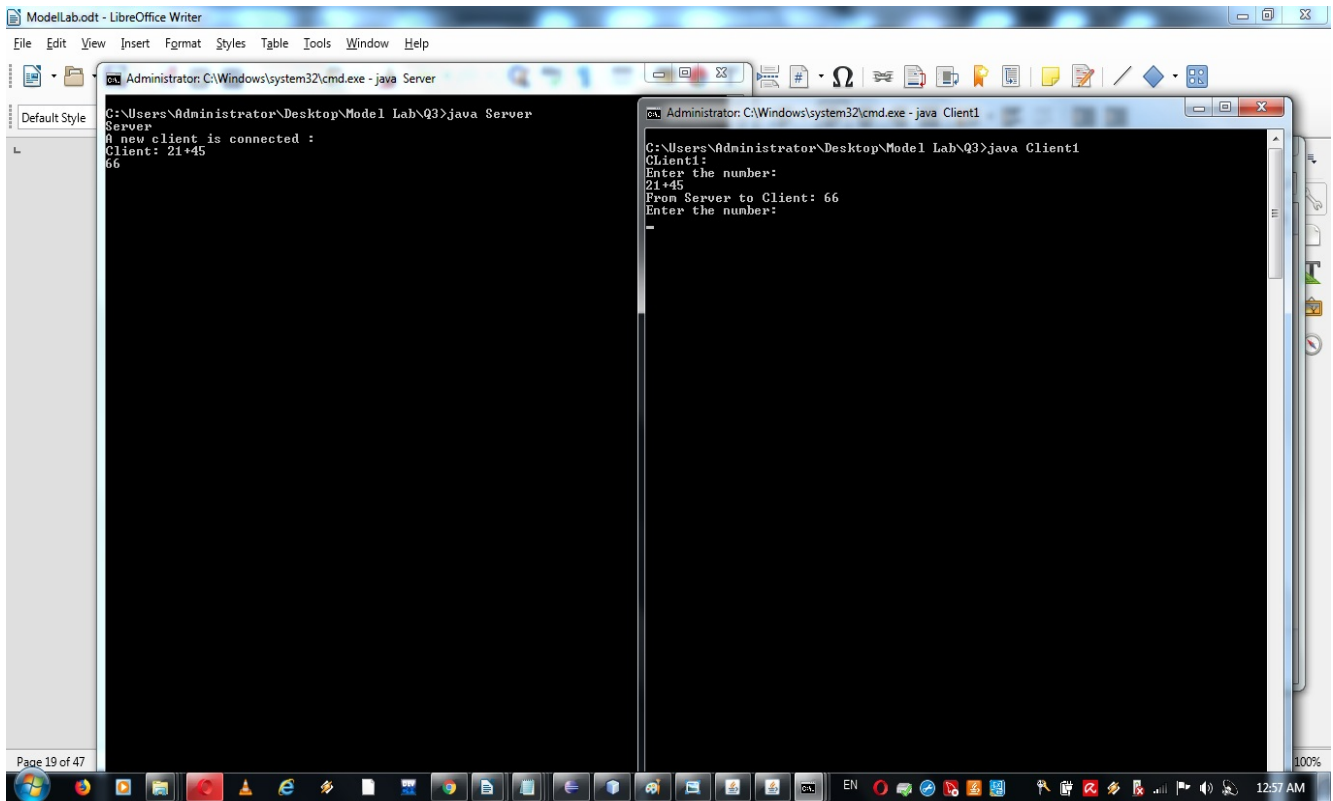
System.out.println("From Server to Client:-" + data(receive));

// break the loop if user enters "bye"
if (inp.equals("bye"))
    break;

}
}
public static StringBuilder data(byte[] a)
{
    if (a == null)
        return null;
    StringBuilder ret = new StringBuilder();
    int i = 0;
    while (a[i] != 0)
    {
        ret.append((char) a[i]);
        i++;
    }
    return ret;
}
}

```

Output:



Exercise 4

Write a java network program in such a way that the server broadcasts quotes at a regular interval and the client passively listens for quotes and does so on a Multicast Socket.

MultiServer.java

/*

- * 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.

```
*/  
package lab6;  
  
/**  
 *  
 * @author 205117066  
 */  
import java.io.IOException;  
import java.net.DatagramPacket;  
import java.net.DatagramSocket;  
import java.net.InetAddress;  
import java.util.Scanner;  
import java.io.*;  
  
public class Multiserver  
{  
  
    public static void main(String[] args) throws IOException  
    {  
        DatagramSocket ds = new DatagramSocket(33066);  
  
        String arr[]=new String[10];  
        Scanner sc=new Scanner(System.in);  
        System.out.println("Enter the message 10");  
        for(int i=0;i<10;i++)  
        {  
            String st=sc.nextLine();  
            arr[i]=st;  
        }  
  
        System.out.println("Server");  
  
        int i=0;  
        while (true)  
        {  
  
            try{  
  
                byte buf[] = null;  
                InetAddress groupip =InetAddress.getByName("230.1.1.1");  
  
                //System.out.println("Enter the quotes");  
                //String inp = sc.nextLine();  
                if(i>=10)  
                    break;  
                buf = arr[i].getBytes();  
                DatagramPacket DpSend = new DatagramPacket(buf, buf.length, groupip, 12345);  
                ds.send(DpSend);  
            }  
        }  
    }  
}
```

```

        Thread.sleep(5000);
        i++;
    }
    catch(Exception e)
    {
        System.out.println(e);
    }
}
}
}

```

MultiClient1.java

```

/*
 * 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.
 */
package lab6;

/**
 *
 * @author 205117066
 */
import java.io.IOException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.net.MulticastSocket;
import java.util.Scanner;
public class Multiclient1 {

    public static void main(String args[]) throws IOException
    {

        System.out.println("Client2");

        DatagramPacket DpReceive = null;
        MulticastSocket ms=new MulticastSocket(12345);
        InetAddress groupip = InetAddress.getByName("230.1.1.1");//224.0.0.1 to 239.255.255.255
        ms.joinGroup(groupip);
        byte [] receive=new byte[65535];
        while(true)
        {

            DpReceive = new DatagramPacket(receive, receive.length);
            ms.receive(DpReceive);

```

```

        System.out.println("New Quotes is: " + data(receive));

        //System.out.println("New Quotes is: " + DpReceive.getData().toString());
        receive = new byte[65535];

    }

}

public static StringBuilder data(byte[] a)
{
    if (a == null)
        return null;
    StringBuilder ret = new StringBuilder();
    int i = 0;
    while (a[i] != 0)
    {
        ret.append((char) a[i]);
        i++;
    }
    return ret;
}
}

```

MultiClient2.java

```

/*
 * 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.
 */
package lab6;

/**
 *
 * @author 205117066
 */
import java.io.IOException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.net.MulticastSocket;
import java.util.Scanner;
public class Multiclient2 {

```

```

    public static void main(String args[]) throws IOException

```

```

{

    System.out.println("Client2");

    DatagramPacket DpReceive = null;
    MulticastSocket ms=new MulticastSocket(12345);
    InetAddress groupip = InetAddress.getByName("230.1.1.1");//224.0.0.1 to 239.255.255.255
    ms.joinGroup(groupip);
    byte [] receive=new byte[65535];
    while(true)
    {

        DpReceive = new DatagramPacket(receive, receive.length);
        ms.receive(DpReceive);
        System.out.println("New Quotes is: " + data(receive));

        //System.out.println("New Quotes is: " + DpReceive.getData().toString());
        receive = new byte[65535];

    }

}

public static StringBuilder data(byte[] a)
{
    if (a == null)
        return null;
    StringBuilder ret = new StringBuilder();
    int i = 0;
    while (a[i] != 0)
    {
        ret.append((char) a[i]);
        i++;
    }
    return ret;
}
}

```

MultiClient3.java

```

/*
 * 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.
 */
package lab6;

/**

```

```

*
* @author 205117066
*/
import java.io.IOException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.net.MulticastSocket;
import java.util.Scanner;
public class Multiclient3 {

    public static void main(String args[]) throws IOException
    {

        System.out.println("Client3");

        DatagramPacket DpReceive = null;
        MulticastSocket ms=new MulticastSocket(12345);
        InetAddress groupip = InetAddress.getByName("230.1.1.1");
        ms.joinGroup(groupip);
        byte [] receive=new byte[65535];
        while(true)
        {

            DpReceive = new DatagramPacket(receive, receive.length);
            ms.receive(DpReceive);
            System.out.println("New Quotes is: " + data(receive));

            receive = new byte[65535];

        }

    }

    public static StringBuilder data(byte[] a)
    {
        if (a == null)
            return null;
        StringBuilder ret = new StringBuilder();
        int i = 0;
        while (a[i] != 0)
        {
            ret.append((char) a[i]);
            i++;
        }
        return ret;
    }
}

```


Output:

```
run:
Enter the message 10
Hello
printre
mouse
pen
apper
mou
monu
nehh
lovey
Raju
Server
```

Client1

```
run:
Client1
New Quotes is: mouse
New Quotes is: pen
New Quotes is: apper
New Quotes is: mou
New Quotes is: monu
New Quotes is: nehh
New Quotes is: lovey
New Quotes is: Raju
```

```
run:
Client2
New Quotes is: pen
New Quotes is: apper
New Quotes is: mou
New Quotes is: monu
New Quotes is: nehh
New Quotes is: lovey
New Quotes is: Raju
```

```
run:
Client3
New Quotes is: monu
New Quotes is: nehh
New Quotes is: lovey
New Quotes is: Raju
```

Exercise 5

Java code for serialization and deserialization of a Java object.

SeriDese.java

```
/*
 * 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.
 */
/**
 *
 * @author 205117066
 */
import java.io.*;
import java.util.Scanner;
class Employee implements Serializable
{

    static int id;
    String fname;
    String lname;
    transient String gender;
    double salary;
    Employee(int id,String fname,String lname,String gender,double salary)
    {

        this.id=id;
        this.lname=lname;
        this.fname=fname;
        this.gender=gender;
        this.salary=salary;
    }
    void Display(Employee obj)
    {
        System.out.println("Id: "+obj.id);
        System.out.println("Name: "+obj.fname+" "+obj.lname);
        System.out.println("Gender: "+obj.gender);
        System.out.println("Salary: "+obj.salary);
    }
}
```

```

public class SeriDese {

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

        Employee object[]=new Employee[10];
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the number of records you want to serialize: ");
        int n=Integer.parseInt(sc.nextLine());
        for(int i=0;i<n;i++)
        {
            System.out.println("Enter the id for employee "+(i+1));
            int id=Integer.parseInt(sc.nextLine());
            System.out.println("Enter the first name");
            String fname=sc.nextLine();
            System.out.println("Enter the last name");
            String lname=sc.nextLine();
            System.out.println("Enter the gender: ");
            String gender=sc.nextLine();
            System.out.println("Enter the salary: ");
            double salary=Double.parseDouble(sc.nextLine());
            object[i]=new Employee(id,fname,lname,gender,salary);

        }
        for(int i=0;i<n;i++)
        {
            object[i].Display(object[i]);
            System.out.println("*****");
        }
        String filename="records.txt";
        try
        {
            FileOutputStream filestream=null;
            ObjectOutputStream ops=null;
            filestream=new FileOutputStream(filename);
            for(int i=0;i<n;i++)
            {

                ops=new ObjectOutputStream(filestream);
                ops.writeObject(object[i]);
            }
            ops.close();
            filestream.close();
            System.out.println("Serialization finished!!");
        }
        catch(Exception e)
        {
            System.out.println(e);
        }
    }
}

```

```

Employee object1[]=new Employee[10];

object[0].id=34;
try
{
    FileInputStream inps;
    ObjectInputStream ois;
    inps=new FileInputStream(filename);
    for(int i=0;i<n;i++)
    {

        ois=new ObjectInputStream(inps);
        object1[i]=(Employee)ois.readObject();

    }
}
catch(Exception e)
{
    System.out.println(e);
}
for(int i=0;i<n;i++)
{
    object1[i].Display(object1[i]);
    System.out.println("*****");
}
}
}

```

Output:

```

C:\Users\Administrator\Desktop\Model Lab\Q5>java SeriDese
Enter the number of records you want to serialize:
2
Enter the id for employee 1
11
Enter the first name
Ram
Enter the last name
Singh
Enter the gender:
Male
Enter the salary:
12334
Enter the id for employee 2
12
Enter the first name
Rani
Enter the last name

```

```

Mukharjee
Enter the gender:
Female
Enter the salary:
12233
Id: 12
Name: Ram Singh
Gender: Male
Salary: 12334.0
*****
Id: 12
Name: Rani Mukharjee
Gender: Female
Salary: 12233.0
*****
Serialization finished!!
Id: 34
Name: Ram Singh
Gender: null
Salary: 12334.0
*****
Id: 34
Name: Rani Mukharjee
Gender: null
Salary: 12233.0
*****

C:\Users\Administrator\Desktop\Model Lab\Q5>

```

Exercise 6

Simple RMI Application to bill the amount purchased by the client.

InterfaceRMI.java

```

import java.rmi.*;
public interface InterfaceRMI extends Remote

```

```
{
    public String [][] getbill(String item[][]) throws RemoteException;
}
```

ImplementInterfaceRMI.java

```
import java.rmi.*;
import java.rmi.server.*;
public class ImplementInterfaceRMI extends UnicastRemoteObject
    implements InterfaceRMI
{
    ImplementInterfaceRMI() throws RemoteException
    {
        super();
    }

    public String [][] getbill(String item[][]) throws RemoteException
    {
        double total=0;
        double tdis=0;
        int i=0;
        int n=item.length;
        String bill[][]=new String[n+1][5];
        while(i<n)
        {
            String pid =item[i][0];
            String pname=item[i][1];
            double price=Double.parseDouble(item[i][2]);
            double dis=5*price/100;
            double netp=price-dis;

            total+=price;
            tdis+=dis;
            bill[i][0]=pid;
            bill[i][1]=pname;
            bill[i][2]=String.valueOf(price);
            bill[i][3]=String.valueOf(dis);
            bill[i][4]=String.valueOf(netp);
            i++;
        }
        bill[i][0]="Total";
        bill[i][1]=String.valueOf(n);
        bill[i][2]=String.valueOf(total);
        bill[i][3]=String.valueOf(tdis);
        bill[i][4]=String.valueOf(total-tdis);
        return bill;
    }
}
```

```
}
```

ServerRMI.java

```
import java.rmi.*;
import java.rmi.registry.*;
public class ServerRMI
{
    public static void main(String args[])
    {
        System.out.println("Server started");
        try
        {
            InterfaceRMI obj = new ImplementInterfaceRMI();

            LocateRegistry.createRegistry(33066);

            Naming.rebind("rmi://localhost:33066"+
                "/bill",obj);
        }
        catch(Exception ae)
        {
            //System.out.println("sr");
            System.out.println("Server"+ae);
        }
    }
}
```

ClientRMI.java

```
import java.rmi.Naming;
import java.sql.Timestamp;
import java.util.Scanner;
class Productlist
{
    int id;
    String pname;
    double price;
    Productlist(int id,String pname,double price)
    {
        this.id=id;
        this.pname=pname;
        this.price=price;
    }
    public void Display()
    {
        System.out.println("Product id: "+id);
        System.out.println("Product name: "+pname);
        System.out.println("Product price: "+price);
    }
}
```

```

}
public class ClientRMI {

    public static void main(String []args)
    {
        Productlist object[]=new Productlist[10];
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the number of products purchased by the client:");
        int n=Integer.parseInt(sc.nextLine());
        for(int i=0;i<n;i++)
        {
            System.out.println("Enter the id for product "+(i+1));
            int id=Integer.parseInt(sc.nextLine());
            System.out.println("Enter the product name");
            String pname=sc.nextLine();

            System.out.println("Enter the price: ");
            double price=Double.parseDouble(sc.nextLine());
            object[i]=new Productlist(id,pname,price);

        }
        String item[][]=new String[n][3];
        System.out.println("Items purchased by client are: ");
        for(int i=0;i<n;i++)
        {
            object[i].Display();
            item[i][0]=String.valueOf(object[i].id);
            item[i][1]=String.valueOf(object[i].pname);
            item[i][2]=String.valueOf(object[i].price);
            System.out.println("*****");
        }
        try
        {

            InterfaceRMI access = (InterfaceRMI)Naming.lookup("rmi://localhost:33066"+ "/bill");
            String Bill[][] = access.getbill(item);
            String tab="\t\t";
            String st="";
            int SN=1001;
            String cname="Client";
            Timestamp timestamp = new Timestamp(System.currentTimeMillis());
            st=st+"S.N.: "+String.valueOf(SN)+tab+"Name: "+cname+"\t"+"Date:
st=st+"*****\n
";

            st=st+"Id\tProduct\tPrice\tDiscount\tNetPrice\n";

```



```

        st=st+"-----\n";

        for(int i=0;i<Bill.length;i++)
        {
            st=st+Bill[i][0]+tab+Bill[i][1]+tab+Bill[i][2]+tab+Bill[i][3]+tab+Bill[i][4]+"\\n";

            if(i==(Bill.length-2))
                st=st+"-----\\n";

        }
        SN++;
        System.out.println("Bill generated!!!");
        System.out.println(st);
    }
    catch(Exception e)
    {
        System.out.println("Client"+e);
    }
}
}

```

Output:

Enter the number of products purchased by the client:

2

Enter the id for product 1

101

Enter the product name

Pen

Enter the price:

10

Enter the id for product 2

102

Enter the product name

pencil

Enter the price:

15

Items purchased by client are:

Product id: 101

Product name: Pen

Product price: 10.0

Product id: 102

Product name: pencil

Product price: 15.0

Bill generated!!!

S.N.: 1001 Name: Client Date: 2019-04-04 22:56:23.559

Id	Product	Price	Discount	NetPrice
101	Pen	10.0	0.5	9.5
102	pencil	15.0	0.75	14.25
Total	2	25.0	1.25	23.75

BUILD SUCCESSFUL (total time: 45 seconds)

Exercise 7

Java RMI with database connectivity (Online Shopping)

RmiInterface.java

```
import java.rmi.*;
import java.util.Set;
public interface RmiInterface extends Remote
{
    public String [][] query(Set<String> cart) throws RemoteException;
    public String [][] GetRs(String table) throws RemoteException;
}
```

RmiImplement.java

```
import java.rmi.*;
import java.rmi.server.*;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import java.util.Set;
```

```
/**
```

```
*
```

```
* @author Administrator
```

```

*/
public class RmiImplement extends UnicastRemoteObject
    implements RmiInterface
{
    RmiImplement() throws RemoteException
    {
        super();
    }

    public String [][] query(Set<String> cart) throws RemoteException
    {

        int n=cart.size();
        String bill[][]=new String[n+1][5];
        try
        {
            Connection con=null;
            Statement st=null;
            ResultSet rs=null;
            String query=null;
            Class.forName("org.sqlite.JDBC");
            con=DriverManager.getConnection("jdbc:sqlite:product.sqlite");
            st=con.createStatement();
            double total=0;
            double tdis=0;
            int i=0;
            for(String val:cart)
            {
                query="SELECT *from productlist where product_id='"+val+"'";
                rs=st.executeQuery(query);
                while(rs.next())
                {
                    String pid =rs.getString(1);
                    String pname=rs.getString(2);
                    double price=rs.getDouble(4);
                    double dis=5*price/100;
                    double netp=price-dis;

                    total+=price;
                    tdis+=dis;
                    bill[i][0]=pid;
                    bill[i][1]=pname;
                    bill[i][2]=String.valueOf(price);
                    bill[i][3]=String.valueOf(dis);
                    bill[i][4]=String.valueOf(netp);
                    i++;

                }

            }
        }
    }
}

```

```

    }
    bill[i][0]="Total";
    bill[i][1]=String.valueOf(n);
    bill[i][2]=String.valueOf(total);
    bill[i][3]=String.valueOf(tdis);
    bill[i][4]=String.valueOf(total-tdis);
    return bill;
}
catch(Exception e)
{
    System.out.println(e);
}
return bill;
}
public String [][] GetRs(String table) throws RemoteException
{
    String[][] lis1=new String[0][4];
    try{

        Class.forName("org.sqlite.JDBC");
        Connection con = DriverManager.getConnection("jdbc:sqlite:product.sqlite"); //To create
database if not exist
        String query = "select *from "+table;
        Statement st = con.createStatement();
        ResultSet rs = st.executeQuery(query);
        int n=0;
        while(rs.next())
        {
            n++;
        }

        //System.out.println(n);
        String[][] list=new String[n][4];
        int i=0;
        rs=st.executeQuery(query);
        while(rs.next())
        {
            list[i][0]=rs.getString(1);
            list[i][1]=rs.getString(2);
            list[i][2]=rs.getString(3);
            list[i][3]=rs.getString(4);
            i++;
        }
        //System.out.println("try");
        return list;
    }
    catch(Exception e)
    {
        System.out.println("IS");
    }
}

```

```

        System.out.println(e);
    }
    System.out.println("out try");
    return lis1;
}
}

```

ImplementServer.java

```

import java.rmi.*;
import java.rmi.registry.*;
public class ImplementServer
{
    public static void main(String args[])
    {
        try
        {
            RmiInterface obj = new RmiImplement();

            LocateRegistry.createRegistry(33066);

            Naming.rebind("rmi://localhost:33066"+
                "/bill",obj);
            RmiInterface obj1 = new RmiImplement();
            Naming.rebind("rmi://localhost:33066"+
                "/database",obj1);

        }
        catch(Exception ae)
        {
            //System.out.println("sr");
            System.out.println(ae);
        }
    }
}

```

Home.java

```

/*
 * 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.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import javax.swing.JOptionPane;

/**
 *
 * @author Administrator
 */
public class Home extends javax.swing.JFrame {

    /**
     * Creates new form Home
     */
    Connection con=null;
    Statement st=null;
    ResultSet rs=null;
    String query=null;

    public Home() {
        initComponents();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        jLabel3 = new javax.swing.JLabel();
        username_field = new javax.swing.JTextField();
        password_field = new javax.swing.JPasswordField();
        login_btn = new javax.swing.JButton();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

        jLabel1.setFont(new java.awt.Font("Tahoma", 0, 24)); // NOI18N
        jLabel1.setForeground(new java.awt.Color(0, 102, 102));
        jLabel1.setText("Enjoy Online Shopping...");
        getContentPane().add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(210, 10,
260, 30));

        jLabel2.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N

```

```

jLabel2.setForeground(new java.awt.Color(0, 102, 102));
jLabel2.setText("Password");
getContentPane().add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(130, 180,
80, 20));

jLabel3.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
jLabel3.setForeground(new java.awt.Color(0, 102, 102));
jLabel3.setText("Username");
getContentPane().add(jLabel3, new org.netbeans.lib.awtextra.AbsoluteConstraints(130, 120,
80, 30));

username_field.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
username_field.setForeground(new java.awt.Color(0, 102, 102));
getContentPane().add(username_field, new
org.netbeans.lib.awtextra.AbsoluteConstraints(220, 120, 220, 30));

password_field.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
password_field.setForeground(new java.awt.Color(0, 102, 102));
getContentPane().add(password_field, new org.netbeans.lib.awtextra.AbsoluteConstraints(220,
170, 220, 30));

login_btn.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
login_btn.setForeground(new java.awt.Color(0, 102, 102));
login_btn.setText("Login");
login_btn.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        login_btnActionPerformed(evt);
    }
});
getContentPane().add(login_btn, new org.netbeans.lib.awtextra.AbsoluteConstraints(280, 250,
70, -1));

pack();
} // </editor-fold>

private void login_btnActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    String user_name=username_field.getText().toString();
    String pass_word=password_field.getText().toString();
    try
    {
        Class.forName("org.sqlite.JDBC");
        con=DriverManager.getConnection("jdbc:sqlite:Consumer.sqlite");//To create database if
not exist
        String query = "select *from CUSTOMER where username='"+user_name+"' and
password='"+pass_word+"'";
        st = con.createStatement();
        rs = st.executeQuery(query);
        if(!rs.next())
        {

```

```

        JOptionPane.showMessageDialog(null,"your username or password is incorrect","Error
Message",JOptionPane.WARNING_MESSAGE);
        System.out.println("Wrong password");
    }
    else
    {
        rs=st.executeQuery(query);

        //JOptionPane.showMessageDialog(null,"Login
sucessfull","Message",JOptionPane.INFORMATION_MESSAGE);
        Product obj=new Product();
        if(rs.next())
        obj.main(rs.getString(1),rs.getString(3));
        dispose();
    }

}
catch(Exception e)
{
    System.out.println(e);
}

}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
    * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
    */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(Home.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    } catch (InstantiationException ex) {

```



```

java.util.logging.Logger.getLogger(Home.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Home.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Home.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    }
//</editor-fold>

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new Home().setVisible(true);
        }
    });
}

// Variables declaration - do not modify
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JButton login_btn;
private javax.swing.JPasswordField password_field;
private javax.swing.JTextField username_field;
// End of variables declaration
}

```

Product.java

```

/*
 * 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.rmi.Naming;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import java.sql.Timestamp;
import java.util.HashSet;
import java.util.Set;
import javax.swing.table.DefaultTableModel;
import javax.swing.table.TableModel;

```

```

/**
 *
 * @author Administrator
 */
public class Product extends javax.swing.JFrame {

    /**
     * Creates new form Product
     */
    Connection con=null;
    Statement st=null;
    ResultSet rs=null;
    String query=null;
    Set<String> cart = new HashSet<String>();
    static String username="205117066";
    static String cname="Surendra";
    int SN=1001;

    public Product() {
        initComponents();
        try
        {
            DefaultTableModel model=(DefaultTableModel)product_table.getModel();
            Object [] row=new Object[4];
            RmiInterface access = (RmiInterface)Naming.lookup("rmi://localhost:33066"+ "/database");
            String list[][]=access.GetRs("productlist");
            //System.out.println(list.length);
            for(int i=0;i<list.length;i++)
            {
                row[0]=list[i][0];
                row[1]=list[i][1];
                row[2]=list[i][2];
                row[3]=list[i][3];
                model.addRow(row);
            }
        }
        catch(Exception e)
        {
            System.out.println("Pr");
            System.out.println(e);
        }
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")

```

```

// <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN:initComponents
private void initComponents() {

    jLabel1 = new javax.swing.JLabel();
    dis_label = new javax.swing.JLabel();
    jButton1 = new javax.swing.JButton();
    jScrollPane1 = new javax.swing.JScrollPane();
    product_table = new javax.swing.JTable();
    dis_label1 = new javax.swing.JLabel();
    jScrollPane3 = new javax.swing.JScrollPane();
    display_table = new javax.swing.JTable();
    dis_label2 = new javax.swing.JLabel();
    jScrollPane2 = new javax.swing.JScrollPane();
    bill_field = new javax.swing.JTextArea();
    bill_label = new javax.swing.JLabel();

    jLabel1.setText("jLabel1");

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
    getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

    dis_label.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
    dis_label.setForeground(new java.awt.Color(0, 153, 153));
    dis_label.setText("To delete item from cart click on product id");
    getContentPane().add(dis_label, new org.netbeans.lib.awtextra.AbsoluteConstraints(480, 260,
250, 20));

    jButton1.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
    jButton1.setForeground(new java.awt.Color(0, 102, 102));
    jButton1.setText("Generate Bill");
    jButton1.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButton1ActionPerformed(evt);
        }
    });
    getContentPane().add(jButton1, new org.netbeans.lib.awtextra.AbsoluteConstraints(640, 310,
-1, -1));

    product_table.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
    product_table.setForeground(new java.awt.Color(0, 102, 102));
    product_table.setModel(new javax.swing.table.DefaultTableModel(
        new Object [][] {

            },
        new String [] {
            "product id", "Product name", "Company", "Price"
        }
    ) {
        boolean[] canEdit = new boolean [] {
            false, false, false, false

```

```

};

public boolean isCellEditable(int rowIndex, int columnIndex) {
    return canEdit [columnIndex];
}
});
product_table.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        product_tableMouseClicked(evt);
    }
});
jScrollPane1.setViewportView(product_table);

//l-r,t-b,
getContentPane().add(jScrollPane1, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 70,
-1, 270));

dis_label1.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
dis_label1.setForeground(new java.awt.Color(0, 153, 153));
dis_label1.setText("Click on item id to add into cart"); //l-r,t-b,len,wid
getContentPane().add(dis_label1, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 40,
210, 20));

display_table.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
display_table.setForeground(new java.awt.Color(0, 102, 102));
display_table.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {

        },
        new String [] {
            "Product id", "Product", "Company", "Price"
        }
    ) {
        boolean[] canEdit = new boolean [] {
            false, false, false, false
        };

        public boolean isCellEditable(int rowIndex, int columnIndex) {
            return canEdit [columnIndex];
        }
    });
display_table.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        display_tableMouseClicked(evt);
    }
});
jScrollPane3.setViewportView(display_table);

getContentPane().add(jScrollPane3, new org.netbeans.lib.awtextra.AbsoluteConstraints(480,
70, 390, 180));

```

```

dis_label2.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
dis_label2.setForeground(new java.awt.Color(0, 153, 153));
dis_label2.setText("Items in Cart");
getContentPane().add(dis_label2, new org.netbeans.lib.awtextra.AbsoluteConstraints(490, 50,
210, 20));

bill_field.setEditable(false);
bill_field.setColumns(20);
bill_field.setFont(new java.awt.Font("Monospaced", 0, 14)); // NOI18N
bill_field.setForeground(new java.awt.Color(0, 153, 51));
bill_field.setRows(5);
jScrollPane2.setViewportView(bill_field);

getContentPane().add(jScrollPane2, new org.netbeans.lib.awtextra.AbsoluteConstraints(20,
370, 590, 300));

bill_label.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
bill_label.setForeground(new java.awt.Color(0, 153, 153));
getContentPane().add(bill_label, new org.netbeans.lib.awtextra.AbsoluteConstraints(84, 380,
110, 20));

pack();
} // </editor-fold> // GEN-END: initComponents

private void product_tableMouseClicked(java.awt.event.MouseEvent evt) { // GEN-
FIRST:event_product_tableMouseClicked
    // TODO add your handling code here:
    int i=product_table.getSelectedRow();
    int j=product_table.getSelectedColumn();

    TableModel ml=product_table.getModel();
    String value=ml.getValueAt(i,j).toString();

    if(!cart.contains(value))
        cart.add(value);
    //System.out.println(cart.size());
    Display_items();

} // GEN-LAST:event_product_tableMouseClicked

private void display_tableMouseClicked(java.awt.event.MouseEvent evt) { // GEN-
FIRST:event_display_tableMouseClicked
    // TODO add your handling code here:

    int i=display_table.getSelectedRow();
    int j=display_table.getSelectedColumn();

    TableModel ml=display_table.getModel();
    String value=ml.getValueAt(i,j).toString();
    cart.remove(value);

```

```

Display_items();

} //GEN-LAST:event_display_tableMouseClicked

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_jButton1ActionPerformed

    try
    {
        RmiInterface access = (RmiInterface)Naming.lookup("rmi://localhost:33066" + "/bill");
        String Bill[][] = access.query(cart);
        String tab="\t\t";
        String st="";
        Timestamp timestamp = new Timestamp(System.currentTimeMillis());
        st=st+"S.N.: "+String.valueOf(SN)+tab+"Name: "+cname+"\t"+"Date: "+String.valueOf(timestamp)+"\n";

st=st+"*****\n";

        st=st+"Id\t\tProduct\t\tPrice\t\tDiscount\t\tNetPrice\n";
        st=st+"-----\n";

        for(int i=0;i<Bill.length;i++)
        {
            st=st+Bill[i][0]+tab+Bill[i][1]+tab+Bill[i][2]+tab+Bill[i][3]+tab+Bill[i][4]+"\\n";

            if(i==(Bill.length-2))
            st=st+"-----\\n";
            //System.out.println(Bill[i][0]+" "+Bill[i][1]+" "+Bill[i][2]+" "+Bill[i][3]+" "+Bill[i][4]);
        }
        SN++;
        cart.removeAll(cart);
        Display_items();
        //st=st+"<html/>";
        bill_field.setText("");

        bill_field.setText(st);
    }
    catch(Exception e)
    {
        System.out.println(e);
    }

} //GEN-LAST:event_jButton1ActionPerformed

public void Display_items()
{
    try
    {

```

```

Class.forName("org.sqlite.JDBC");
con=DriverManager.getConnection("jdbc:sqlite:product.sqlite");
st=con.createStatement();

DefaultTableModel model1=(DefaultTableModel)display_table.getModel();

Object [] row1=new Object[4];
model1.setRowCount(0);

for(String val:cart)
{
    query="SELECT *from productlist where product_id='"+val+"'";
    rs=st.executeQuery(query);

    while(rs.next())
    {
        row1[0]=rs.getString(1);
        row1[1]=rs.getString(2);
        row1[2]=rs.getString(3);
        row1[3]=rs.getString(4);
        model1.addRow(row1);
    }
}

}
catch(Exception e)
{
    System.out.println(e);
}
}
/**
 * @param args the command line arguments
 */
public static void main(String usern,String name) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
     * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
     */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {

```

```

java.util.logging.Logger.getLogger(Product.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Product.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Product.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

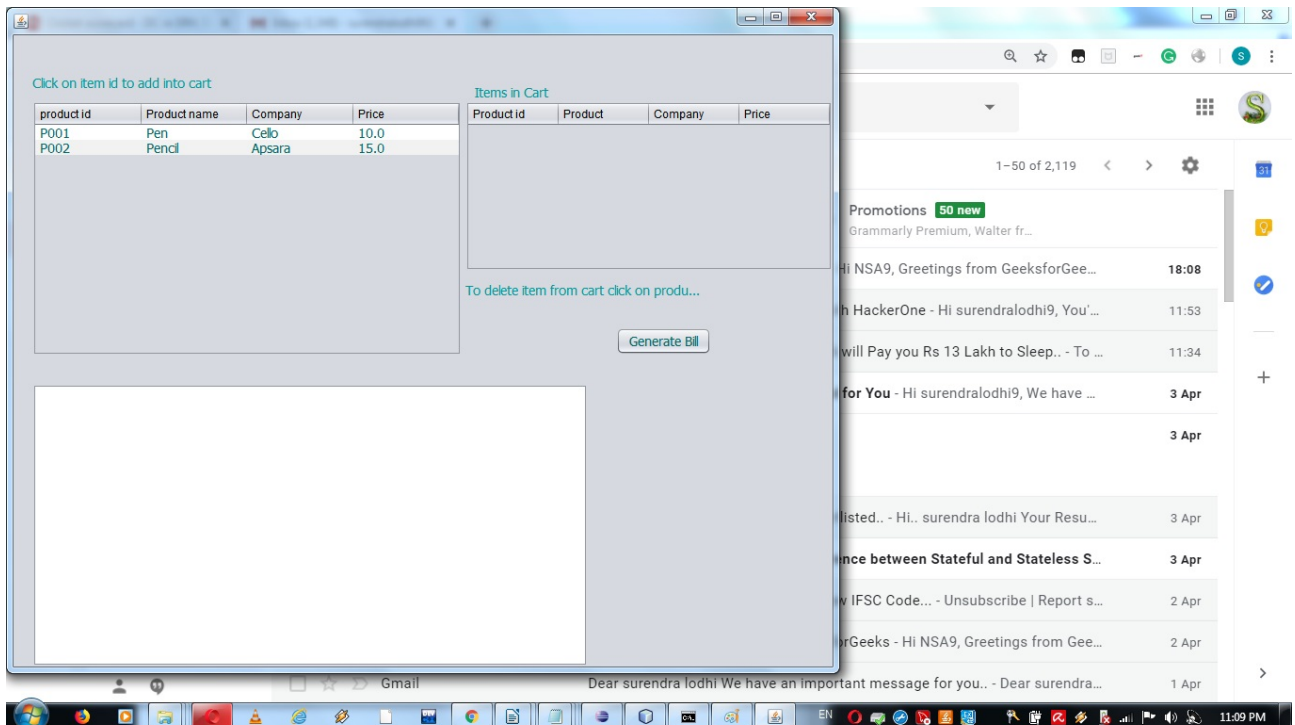
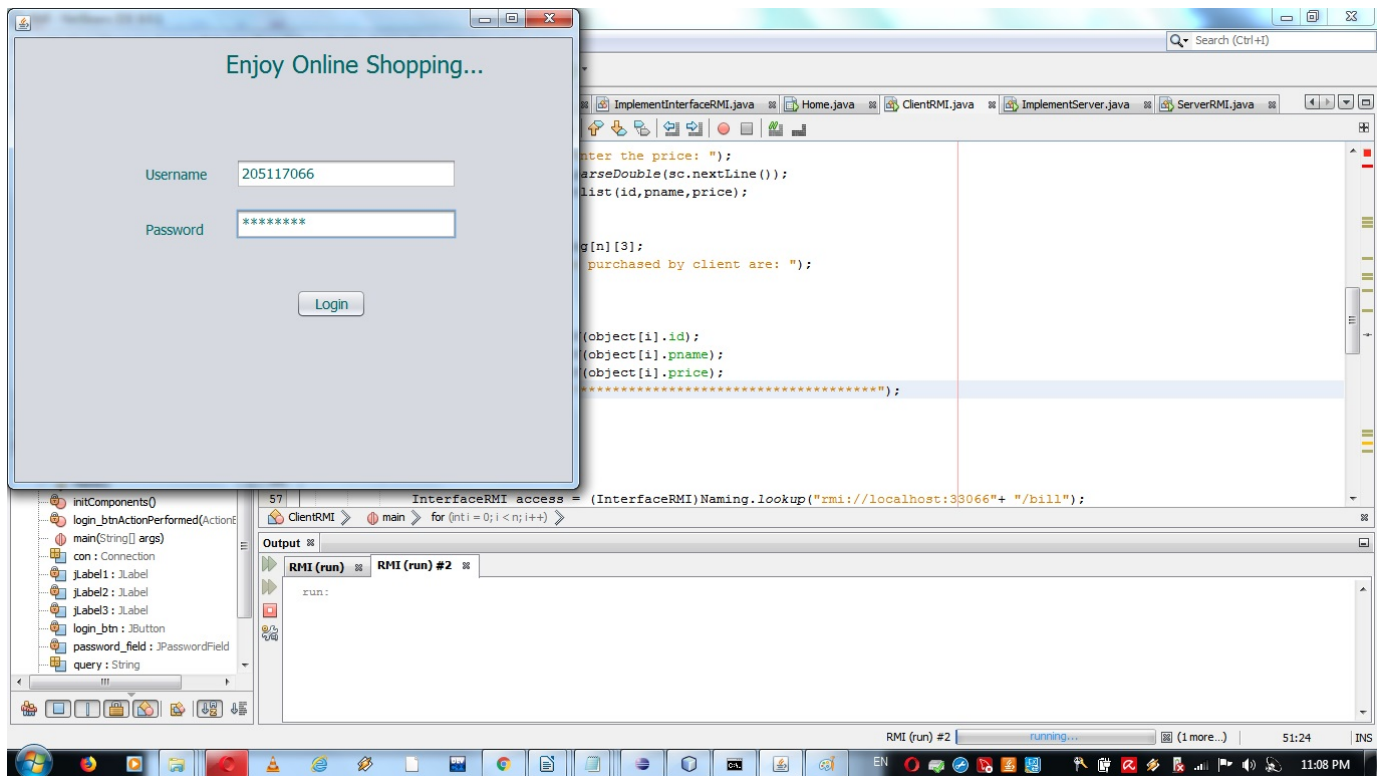
java.util.logging.Logger.getLogger(Product.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    }
//</editor-fold>

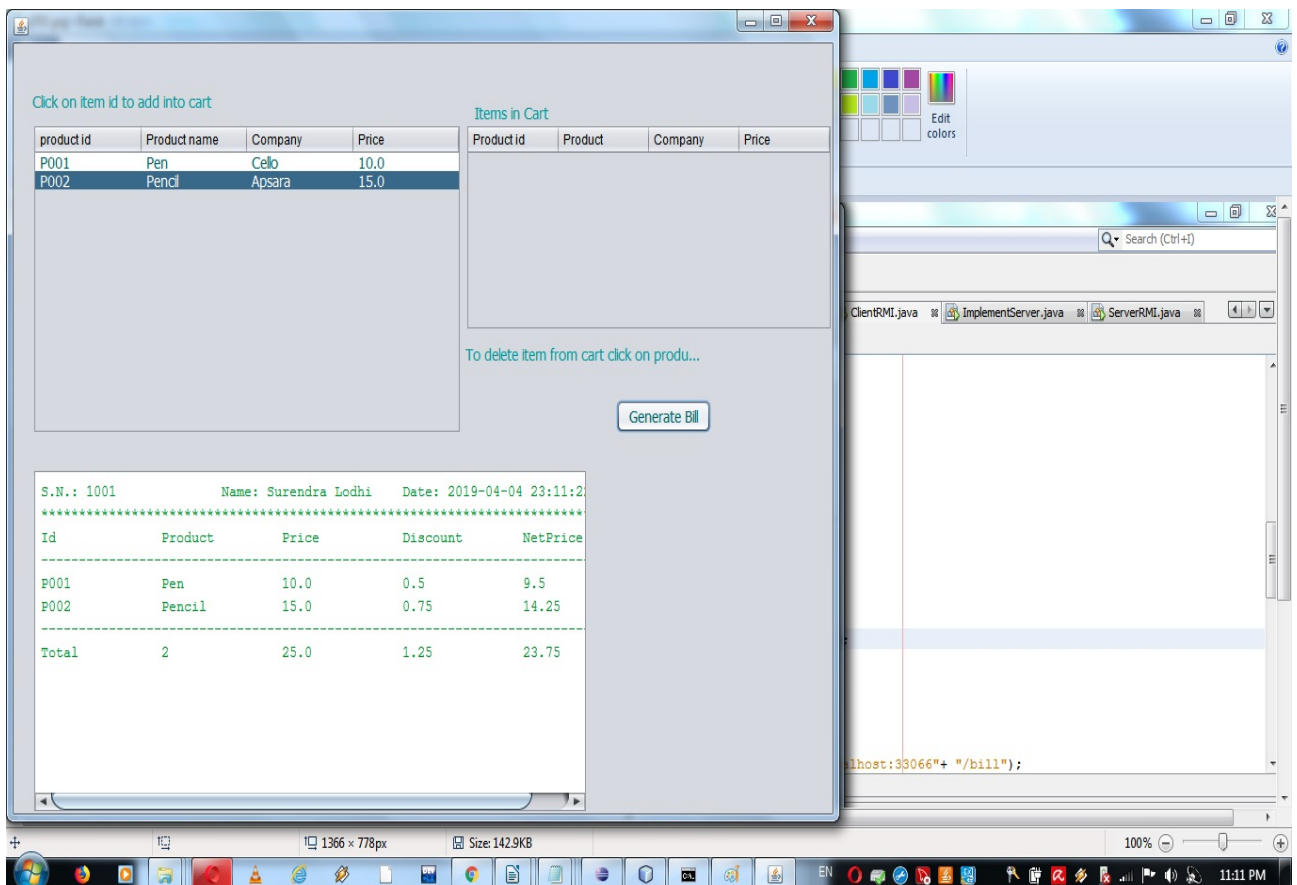
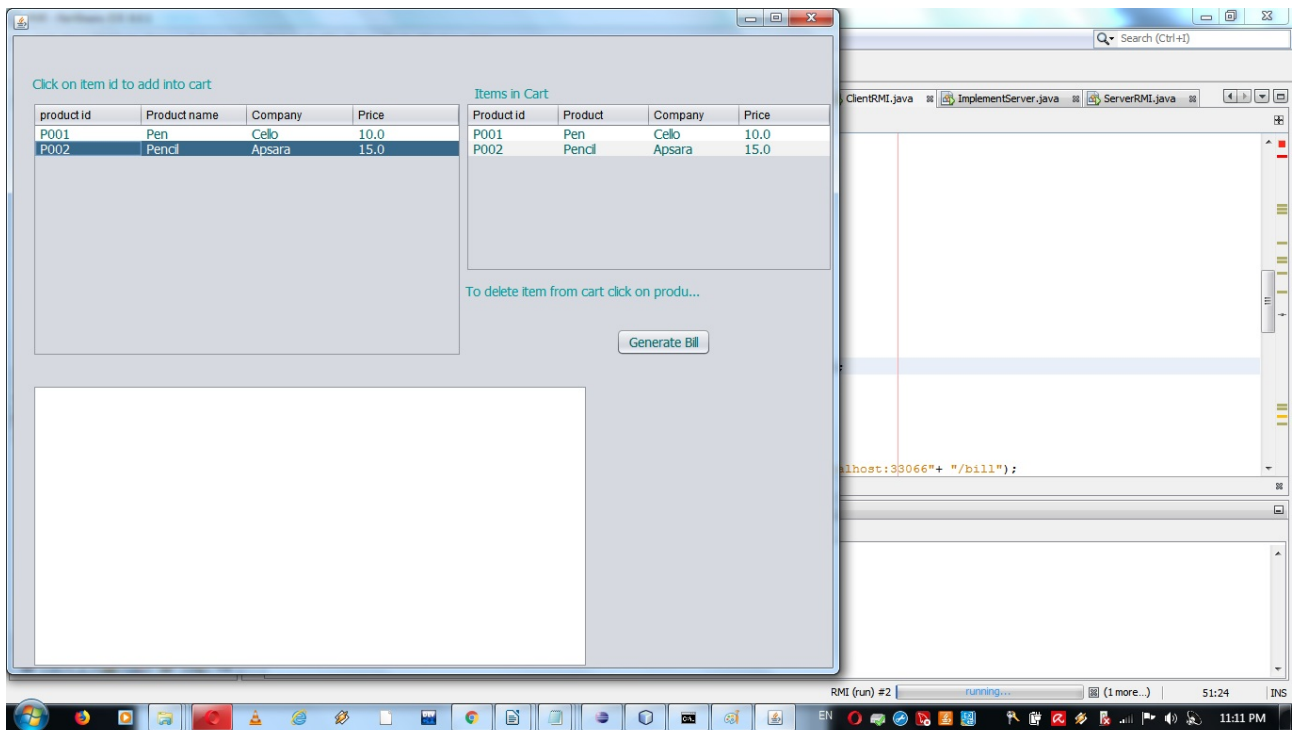
/* Create and display the form */
username=usern;
cname=name;
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        new Product().setVisible(true);
    }
});
}

// Variables declaration - do not modify//GEN-BEGIN:variables
private javax.swing.JTextArea bill_field;
private javax.swing.JLabel bill_label;
private javax.swing.JLabel dis_label;
private javax.swing.JLabel dis_label1;
private javax.swing.JLabel dis_label2;
private javax.swing.JTable display_table;
private javax.swing.JButton jButton1;
private javax.swing.JLabel jLabel1;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JScrollPane jScrollPane2;
private javax.swing.JScrollPane jScrollPane3;
private javax.swing.JTable product_table;
// End of variables declaration//GEN-END:variables
}

```

Output:





Exercise 8

Create a web service using JAX-WS for Banking Application.

```
//creating wb service using netbeans
/*
 * 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.
 */
package org.me.bank;

import javax.jws.WebService;
import javax.jws.WebMethod;
import javax.jws.WebParam;

/**
 *
 * @author 205117080
 */
@WebService(serviceName = "BankServer")
public class BankServer {

    private Integer bal=0;
    private String acc="1241244";

    /**
     * Web service operation
     */
    @WebMethod(operationName = "showBalance")

    public String showBalance()
    {
        //TODO write your implementation code here:
        return bal.toString();
    }

    /**
     * Web service operation
     */
    @WebMethod(operationName = "Deposit")
    public String Deposit(@WebParam(name = "parameter") String parameter) {
        //TODO write your implementation code here:

        if(bal>0)
```

```

        bal=bal+Integer.parseInt(parameter);
System.out.println("error");
        return null;
    }

    /**
     * Web service operation
     */

    @WebMethod(operationName = "Withdraw")
    public String Withdraw(@WebParam(name = "parameter") String parameter) {
        //TODO write your implementation code here:
        if(Integer.parseInt(parameter)<bal)
        {
            bal=bal-Integer.parseInt(parameter);
            return null;
        }
        else
        {
            return "balance is not sufficient";
        }
    }

    /**
     * Web service operation
     */

    @WebMethod(operationName = "Account_no")
    public String Account_no() {
        //TODO write your implementation code here:
        return acc;
    }
}

//A java application for consume the web Service.
/*
 * 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.
 */
package cust;

import java.util.Scanner;

/**
 *
 * @author dhirendra gurjar
 */
public class Cust {

```

```

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

    Scanner sc = new Scanner(System.in);

    while (true) {

        System.out.println("1.Account no.\n" + "2.Deposite\n" + "3.Withdraw\n" + "4.Balance\n");
        System.out.println("enter your choice:");
        int ch = sc.nextInt();
        switch (ch) {
            case 1: {
                try {
                    System.out.println(accountNo());
                } catch (Exception e) {

                }
                break;
            }
            case 2: {
                try {

                    System.out.println("enter the ammount");
                    String bal = sc.next();

                    System.out.println(deposit(bal));
                } catch (Exception e) {

                }
                break;
            }
            case 3: {

                try {

                    System.out.println("enter the ammount");
                    String bal = sc.next();
                    if (withdraw(bal) != null) {
                        System.out.println("Sorry! Insufficient balance");
                    }
                } catch (Exception e) {

                }
                break;
            }
            case 4: {

```

```

        try {

            System.out.println(showBalance());
        } catch (Exception e) {

        }
        break;
    }
}

}

}

private static String accountNo() {
    org.me.bank.BankServer_Service service = new org.me.bank.BankServer_Service();
    org.me.bank.BankServer port = service.getBankServerPort();
    return port.accountNo();
}

private static String deposit(java.lang.String parameter) {
    org.me.bank.BankServer_Service service = new org.me.bank.BankServer_Service();
    org.me.bank.BankServer port = service.getBankServerPort();
    return port.deposit(parameter);
}

private static String withdraw(java.lang.String parameter) {
    org.me.bank.BankServer_Service service = new org.me.bank.BankServer_Service();
    org.me.bank.BankServer port = service.getBankServerPort();
    return port.withdraw(parameter);
}

private static String showBalance() {
    org.me.bank.BankServer_Service service = new org.me.bank.BankServer_Service();
    org.me.bank.BankServer port = service.getBankServerPort();
    return port.showBalance();
} }

```

Output:

Testing the webservice

Method invocation trace x

showBalance Method invocation

Method parameter(s)

Type	Value
java.lang.String	"800"

Method returned

java.lang.String : "800"

SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <S:Body>
    <ns2:showBalance xmlns:ns2="http://bank.me.org/" />
  </S:Body>
</S:Envelope>
```

SOAP Response

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <S:Body>
    <ns2:showBalanceResponse xmlns:ns2="http://bank.me.org/">
      <return>800</return>
    </ns2:showBalanceResponse>
  </S:Body>
</S:Envelope>
```

BankServer Web Service Tester

This form will allow you to test your web service implementation ([WSDL File](#))

To invoke an operation, fill the method parameter(s) input boxes and click on the button labeled with the method name.

Methods :

public abstract java.lang.String org.me.bank.BankServer.withdraw(java.lang.String)

public abstract java.lang.String org.me.bank.BankServer.accountNo()

public abstract java.lang.String org.me.bank.BankServer.deposit(java.lang.String)

public abstract java.lang.String org.me.bank.BankServer.showBalance()

//Running On Java Application or Consuming The Service

```
Customer - NetBeans IDE 8.0.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+F)

enter your choice:
1
1241244
1.Account no.
2.Deposite
3.Withdraw
4.Balance

enter your choice:
2
enter the amount
2000
amount deposited
1.Account no.
2.Deposite
3.Withdraw
4.Balance

enter your choice:
4
2000
1.Account no.
2.Deposite
3.Withdraw
4.Balance

enter your choice:
3
enter the amount
1200
1.Account no.
2.Deposite
3.Withdraw
4.Balance

enter your choice:
3
enter the amount
5000
Sorry! Insufficient balance
1.Account no.
2.Deposite
3.Withdraw
```

Customer (run-single) | running... | 41:9 | JMS

Type here to search

08:17
05/04/2019