Practical No. 01

Programs on Remote Process Communication

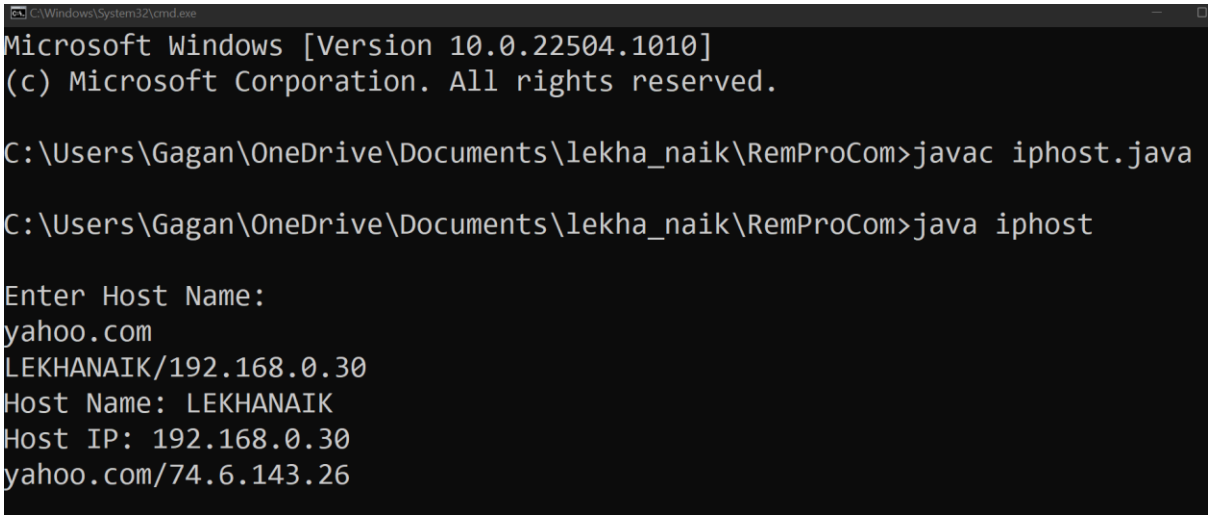
Aim: 1) WAP to find IP address and host information.

Code:

iphost.java

```
import java.net.*;
import java.net.InetAddress;
import java.util.*;
public class iphost
{
        public static void main(String[] args)
        {
                String hostName=null;
                Scanner input=new Scanner(System.in);
                System.out.println("\nEnter Host Name: ");
                hostName=input.nextLine();
        try         {
                InetAddress address=InetAddress.getLocalHost();
                System.out.println(address);
                String name = address.getHostName();
                System.out.println("Host Name: "+name);
                String host = address.getHostAddress();
                System.out.println("Host IP: "+host);
                InetAddress add = InetAddress.getByName(hostName);
                System.out.println(add.toString());
            }
        catch(Exception e)
           {
                System.out.println("Could not find Local Address");
           }    }
}
```

Output:



Practical No. 01

Aim: 2) Write a Socket Program to connect Client and Server through TCP.

Code:

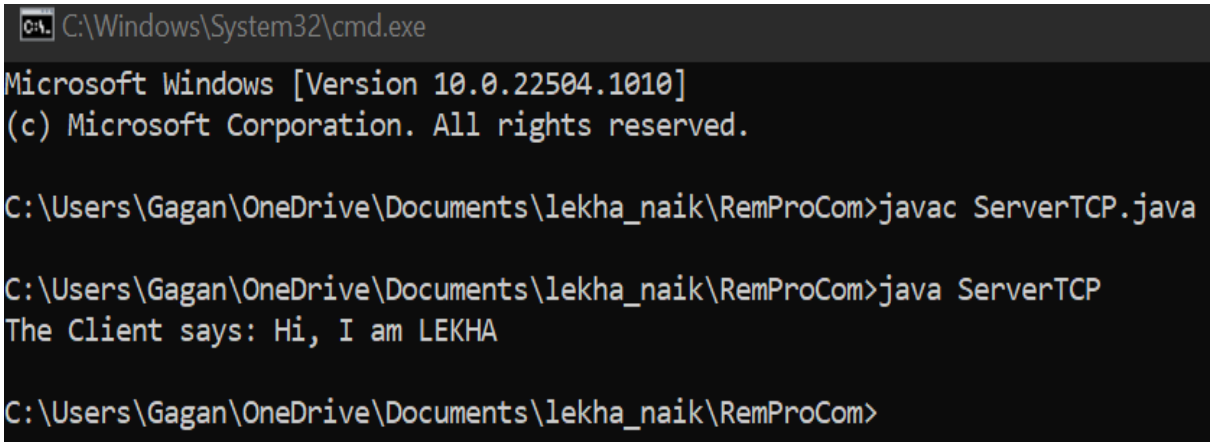
ServerTCP.java

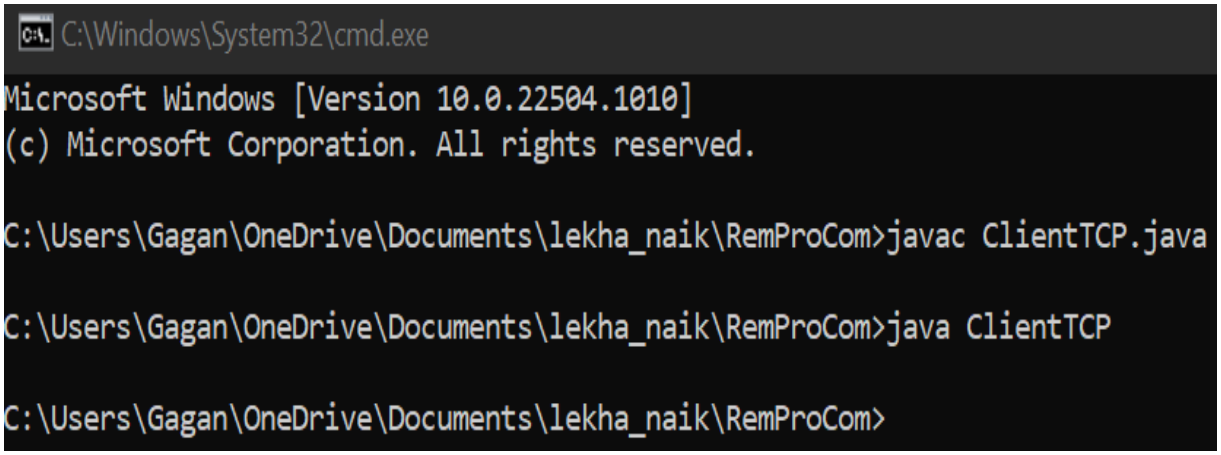
```
import java.net.*;
import java.util.*;
import java.io.*;
public class ServerTCP
{
	public static void main(String[] args)throws Exception
	{
		ServerSocket servSock=new ServerSocket(1234);
		    Socket link=servSock.accept();
		Scanner input=new Scanner(link.getInputStream());
		PrintWriter output =  new PrintWriter(link.getOutputStream());
		String msg=input.nextLine();
		System.out.println("The Client says: "+msg);
	}
}
```

ClientTCP.java

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

public class ClientTCP
{
	public static void main(String[] args)throws Exception
	{
		Socket sock=new Socket("localhost",1234);
		Scanner input = new Scanner(sock.getInputStream());
		PrintWriter output = new PrintWriter(sock.getOutputStream(),true);
		output.println("Hi, I am LEKHA");
		sock.close();
	}
}
```

Output:


C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22504.1010]
(c) Microsoft Corporation. All rights reserved.
C:\Users\Gagan\OneDrive\Documents\lekha_naik\RemProCom>javac ServerTCP.java
C:\Users\Gagan\OneDrive\Documents\lekha_naik\RemProCom>java ServerTCP
The Client says: Hi, I am LEKHA
C:\Users\Gagan\OneDrive\Documents\lekha_naik\RemProCom>


C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22504.1010]
(c) Microsoft Corporation. All rights reserved.
C:\Users\Gagan\OneDrive\Documents\lekha_naik\RemProCom>javac ClientTCP.java
C:\Users\Gagan\OneDrive\Documents\lekha_naik\RemProCom>java ClientTCP
C:\Users\Gagan\OneDrive\Documents\lekha_naik\RemProCom>

Practical No. 01

Aim: 3) Write a Socket Program to connect Client and Server through UDP.

Code:

ServerUDP.java

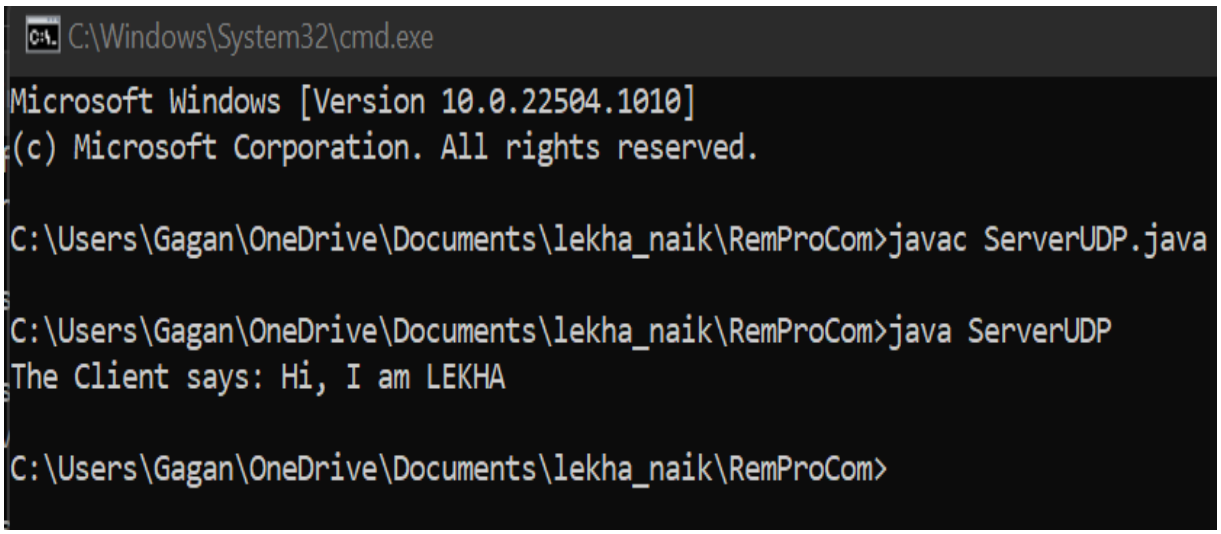
```
import java.net.*;
public class ServerUDP
{
	public static void main(String[] args)throws Exception
 {
		DatagramSocket ds = new DatagramSocket(1234);
		byte[] buffer = new byte[1024];
		DatagramPacket dp = new DatagramPacket(buffer, buffer.length);
		ds.receive(dp);
		String msg = new String(dp.getData(),0,dp.getLength());
		System.out.println("The Client says: "+msg);
		InetAddress clientip = dp.getAddress();
		int clientport = dp.getPort();
		String rep = "Hello, I am NAIK";
		DatagramPacket out = new DatagramPacket(rep.getBytes(), rep.length(), clientip,
clientport);
		ds.send(out);
 }
}
```

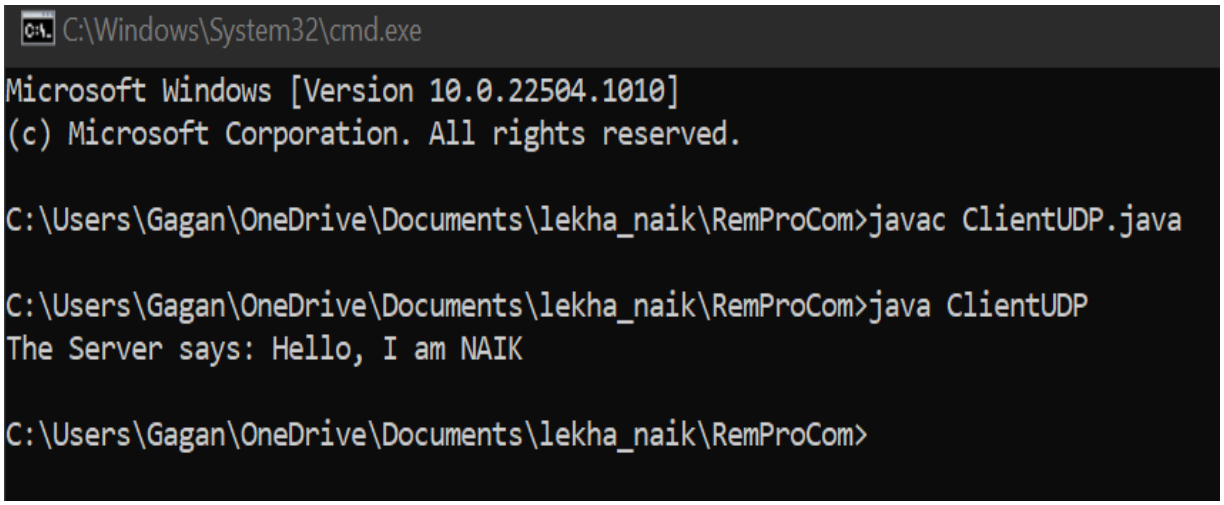
ClientUDP.java

```
import java.net.*;
public class ClientUDP
{
  public static void main(String[] args)throws Exception
		{
		DatagramSocket ds = new DatagramSocket();
		InetAddress ip = InetAddress.getByName("localhost");
		String msg = "Hi, I am LEKHA";
		DatagramPacket sendPacket = new DatagramPacket(msg.getBytes(),msg.length(),ip,1234);
```

```
        ds.send(sendPacket);
        byte[] buffer = new byte[256];
        DatagramPacket inPacket = new DatagramPacket(buffer,buffer.length);
        ds.receive(inPacket);
        String response = new String(inPacket.getData(),0,inPacket.getLength());
        System.out.println("The Server says: "+response);
        }
}
```

Output:





Practical No. 01

Aim: 4) Develop a program for client server chat using java socket.

Code:

ServerChat.java

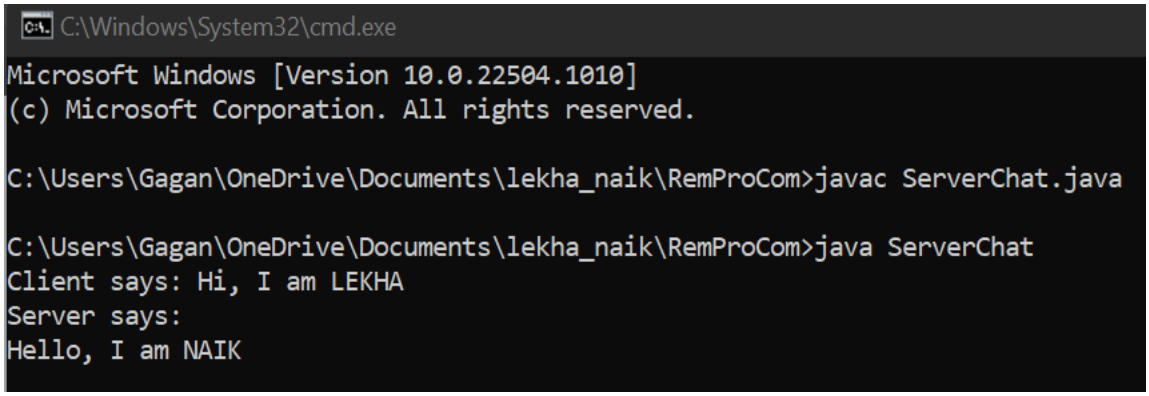
```
import java.net.*;
import java.io.*;
import java.util.*;
public class ServerChat
{
	public static void main(String[] args)throws Exception
    {
		String str;
		ServerSocket ss = new ServerSocket(3333);
		Socket s = ss.accept();
		DataInputStream dis = new DataInputStream(s.getInputStream());
		DataOutputStream dos = new DataOutputStream(s.getOutputStream());
		Scanner input = new Scanner(System.in);
		while(true)
		{
			str = dis.readUTF();
			if(str.equals("exit"))
			{
				dos.writeUTF("exit");
				break;
			}
			System.out.println("Client says: "+str);
			System.out.println("Server says: ");
			str=input.nextLine();
			dos.writeUTF(str);
		}
	ss.close();
	s.close();
     }}
```

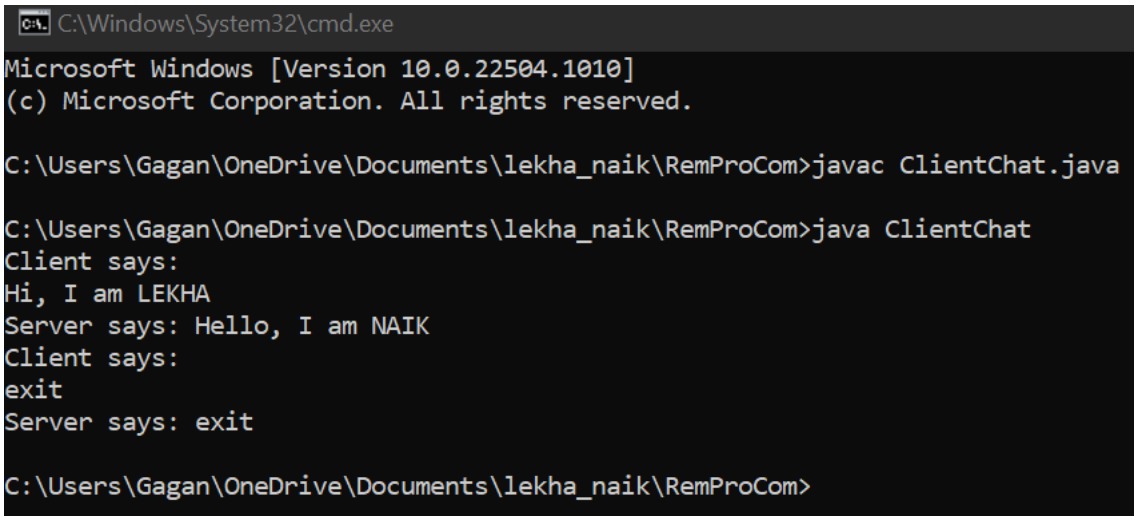
ClientChat.java

```
import java.net.*;

import java.io.*;

import java.util.*;

public class ClientChat

{

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

	{

		String clientMsg, serverMsg;
```

```
            Socket s=new Socket("localhost",3333);
            DataInputStream dis = new DataInputStream(s.getInputStream());
            DataOutputStream dos = new DataOutputStream(s.getOutputStream());
            Scanner input = new Scanner(System.in);
            while(true)
            {
            System.out.println("Client says: ");
            clientMsg = input.nextLine();
            dos.writeUTF(clientMsg);
            serverMsg = dis.readUTF();
            System.out.println("Server says: "+serverMsg);
            if(serverMsg.equals("exit"))
            break;
            }
        s.close();
    }}
```

Output:





Practical No. 01

Aim: 5) Develop a program for client server GUI chat.

Code:

ServerChatGUI.java

```
import java.io.*;
import java.net.*;
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;
import java.util.*;

public class ServerChatGUI extends JFrame implements ActionListener, Runnable {
	JButton b;
	JButton c;
	JTextField tf;
	JTextArea ta;
	ServerSocket ss;
	Socket s;
	PrintWriter pw;
	BufferedReader br;
	Thread th;
	public ServerChatGUI() {
		b = new JButton("Send");
		b.addActionListener(this);
		c = new JButton("Close");
		tf = new JTextField(30);
		ta = new JTextArea(20, 40);
		add(tf);
		add(b);
		add(c);
		add(ta);
		try {
			ss = new ServerSocket(1245);
			s = ss.accept();
```

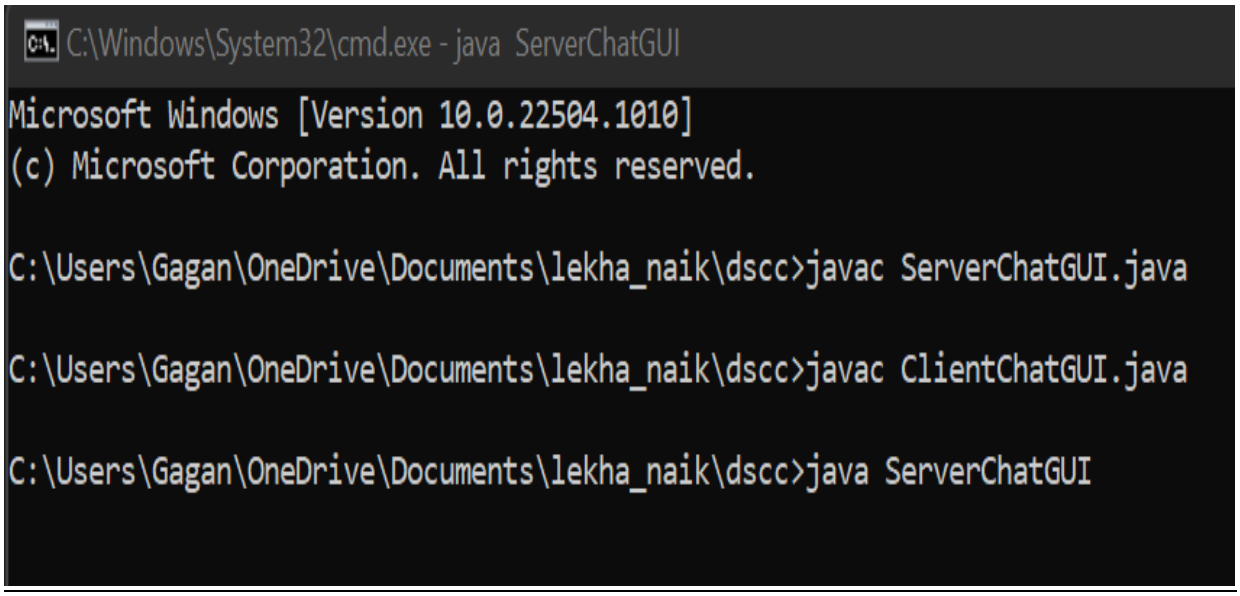
```
                    ta.append("Server Started:" + new Date() + "\n");
                    br = new BufferedReader(new InputStreamReader(s.getInputStream()));
                    pw = new PrintWriter(s.getOutputStream(), true);
                } catch (Exception e) {
                }c.addActionListener(new ActionListener() {
                    public void actionPerformed(ActionEvent e) {
                        pw.println("Client Left");
                        ta.append("You are disconnected now.\n");
                        dispose();
                    }
                });
                th = new Thread(this);
                th.start();
        }
        public void actionPerformed(ActionEvent ae) {
                pw.println(tf.getText());
                ta.append("Server Says:" + tf.getText() + "\n");
                tf.setText(" ");
        }
        public void run() {
                while (true) {
                    try {
                        ta.append("Client says:" + br.readLine() + "\n");
                    } catch (Exception e) {
                    }                    }
        }
        public static void main(String[] args) {
                ServerChatGUI sc = new ServerChatGUI();
                sc.setLayout(new FlowLayout());
                sc.setSize(600, 500);
                sc.setTitle("Server");
                sc.setVisible(true);
        }
}
```

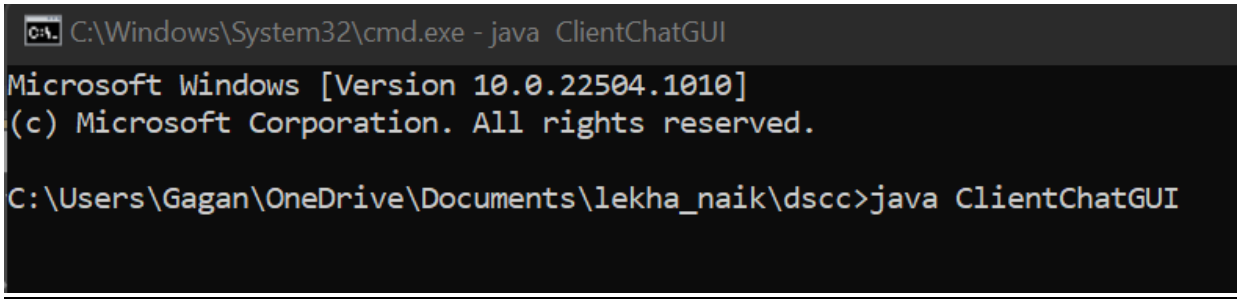
ClientChatGUI.java

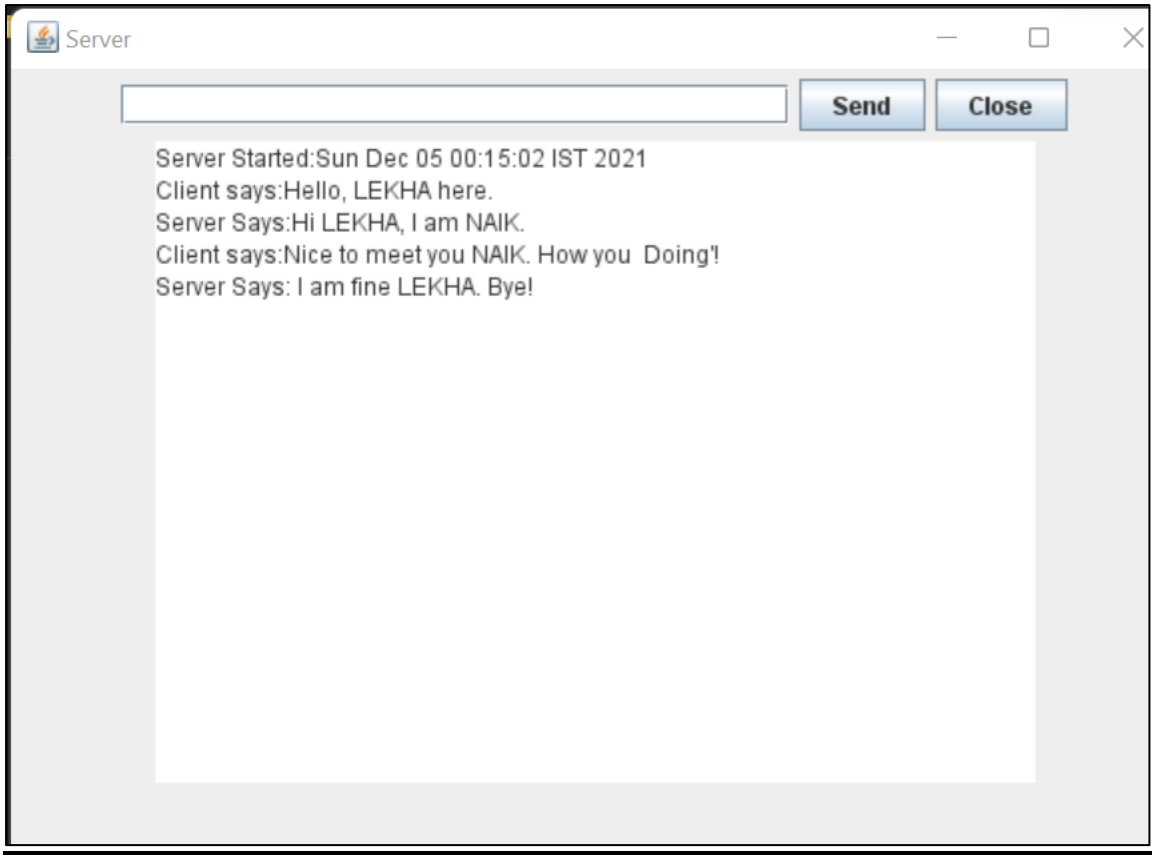
```
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;
import java.io.*;
import java.net.*;

public class ClientChatGUI extends JFrame implements ActionListener, Runnable {
	JButton b;
	JButton c;
	JTextField tf;
	JTextArea ta;
	Socket s;
	PrintWriter pw;
	BufferedReader br;
	Thread th;

	public ClientChatGUI() {
		b = new JButton("Send");
		c = new JButton("Close");
		b.addActionListener(this);
		tf = new JTextField(30);
		ta = new JTextArea(20, 40);
		add(tf);
		add(b);
		add(c);
		add(ta);
		try {
			s = new Socket("localhost", 1245);
			br = new BufferedReader(new InputStreamReader(s.getInputStream()));
			pw = new PrintWriter(s.getOutputStream(), true);
		} catch (Exception e) {
		}
		c.addActionListener(new ActionListener() {
```

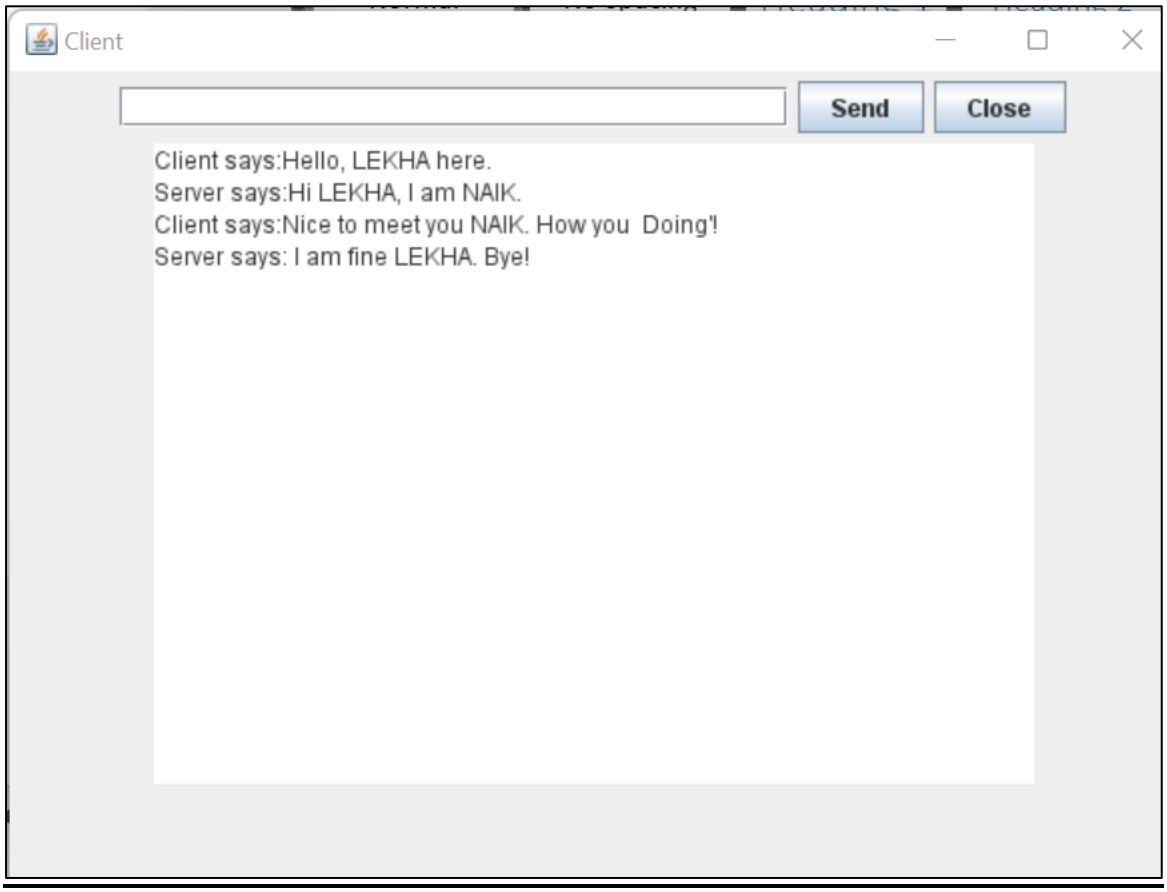
```
                    public void actionPerformed(ActionEvent e) {
                        pw.println("Client Left");
                        ta.append("You are disconnected now.\n");
                        dispose();
                    }
                });
                th = new Thread(this);
                th.start();
        }

        public void actionPerformed(ActionEvent ae) {
                pw.println(tf.getText());
                ta.append("Client says:" + tf.getText() + "\n");
                tf.setText("");
        }

        public void run() {
                while (true) {
                    try {
                        ta.append("Server says:" + br.readLine() + "\n");
                    } catch (Exception e) {
                    }
                }
        }

        public static void main(String[] args) {
                ClientChatGUI c = new ClientChatGUI();
                c.setLayout(new FlowLayout());
                c.setSize(600, 500);
                c.setTitle("Client");
                c.setVisible(true);
        }
}
```

Output:


C:\Windows\System32\cmd.exe - java ServerChatGUI
Microsoft Windows [Version 10.0.22504.1010]
(c) Microsoft Corporation. All rights reserved.
C:\Users\Gagan\OneDrive\Documents\lekha_naik\dscc>javac ServerChatGUI.java
C:\Users\Gagan\OneDrive\Documents\lekha_naik\dscc>javac ClientChatGUI.java
C:\Users\Gagan\OneDrive\Documents\lekha_naik\dscc>java ServerChatGUI


C:\Windows\System32\cmd.exe - java ClientChatGUI
Microsoft Windows [Version 10.0.22504.1010]
(c) Microsoft Corporation. All rights reserved.
C:\Users\Gagan\OneDrive\Documents\lekha_naik\dscc>java ClientChatGUI


Server
Send
Close
Server Started:Sun Dec 05 00:15:02 IST 2021
Client says:Hello, LEKHA here.
Server Says:Hi LEKHA, I am NAIK.
Client says:Nice to meet you NAIK. How you Doing'!
Server Says: I am fine LEKHA. Bye!


Client
Send
Close
Client says:Hello, LEKHA here.
Server says:Hi LEKHA, I am NAIK.
Client says:Nice to meet you NAIK. How you Doing'!
Server says: I am fine LEKHA. Bye!

Practical No. 01

Aim: 6) Implement a server which calculates sum of two numbers using Java Socket.

Code:

ServerAdd.java

```
import java.net.*;
import java.io.*;
public class ServerAdd
{
	private static final int PORT=2222;
	public static void main(String[] args)throws Exception
	{
		ServerSocket srvsoc=new ServerSocket(PORT);
		while(true)
		{
		Socket soc = srvsoc.accept();
		DataInputStream dis=new DataInputStream(soc.getInputStream());
		DataOutputStream dos=new DataOutputStream(soc.getOutputStream());
		int num1 = dis.read();
		int num2= dis.read();
		int add = num1 + num2;
		dos.write(add);
		}	}}
```

ClientAdd.java

```
import java.net.*;

import java.io.*;

import java.util.*;

public class ClientAdd

{

	private static InetAddress host;

	private static final int PORT=2222;

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

	{

		try

		{

			host=InetAddress.getLocalHost();

		}

		catch(UnknownHostException e)

		{
```

```
                System.out.println("Host Not Found");
                System.exit(1);
            }
            int num1,num2;
            Socket soc=new Socket(host,PORT);
            DataInputStream dis=new DataInputStream(soc.getInputStream());
            DataOutputStream dos=new DataOutputStream(soc.getOutputStream());
            Scanner input=new Scanner(System.in);
            System.out.println("Input First Value: ");
            num1=input.nextInt();
            System.out.println("Input Second Value: ");
            num2=input.nextInt();
            dos.write(num1);
            dos.write(num2);
            int add=dis.read();
            System.out.println("Answer is: "+add);
            soc.close();
    }}
```

Output:

```
C:\Windows\System32\cmd.exe - java  ServerAdd
Microsoft Windows [Version 10.0.22504.1010]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Gagan\OneDrive\Documents\lekha_naik\RemProCom>javac ServerAdd.java

C:\Users\Gagan\OneDrive\Documents\lekha_naik\RemProCom>java ServerAdd
```

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22504.1010]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Gagan\OneDrive\Documents\lekha_naik\RemProCom>javac ClientAdd.java

C:\Users\Gagan\OneDrive\Documents\lekha_naik\RemProCom>java ClientAdd
Input First Value:
23
Input Second Value:
34
Answer is: 57
```

Practical No. 01

Aim: 7) Implement a server to find whether an entered number is odd or even using Datagram Socket

Code:

ServerEvOd.java

```
import java.io.IOException;
import java.net.*;
public class ServerEvOd
{
   public static void main(String[] args) throws IOException
   {
      DatagramSocket ds = new DatagramSocket(1234);
      byte[] receive = new byte[65535];
      DatagramPacket DpReceive = null;
      byte buf[] = null;
         DpReceive = new DatagramPacket(receive, receive.length);
         ds.receive(DpReceive);
         String str=new String(DpReceive.getData(),0,DpReceive.getLength());
         System.out.println("Number Entered by Client is " + str);
         String parts[] = str.split(" ", 2);
            int i=Integer.parseInt(parts[0]);
            String s=Integer.toString(i);
            String inp2 ;
            if ( i % 2 == 0 ){
                inp2 =s+" is an Even Number!";
                System.out.println(inp2);}
             else{
                inp2 =s+" is an Odd Number!";
                System.out.println(inp2);}
            String inp = inp2;
            InetAddress ip = DpReceive.getAddress();
            int cp = DpReceive.getPort();
            DatagramPacket DpSend = new DatagramPacket(inp.getBytes(),inp.length(),ip,cp);
            ds.send(DpSend);
```

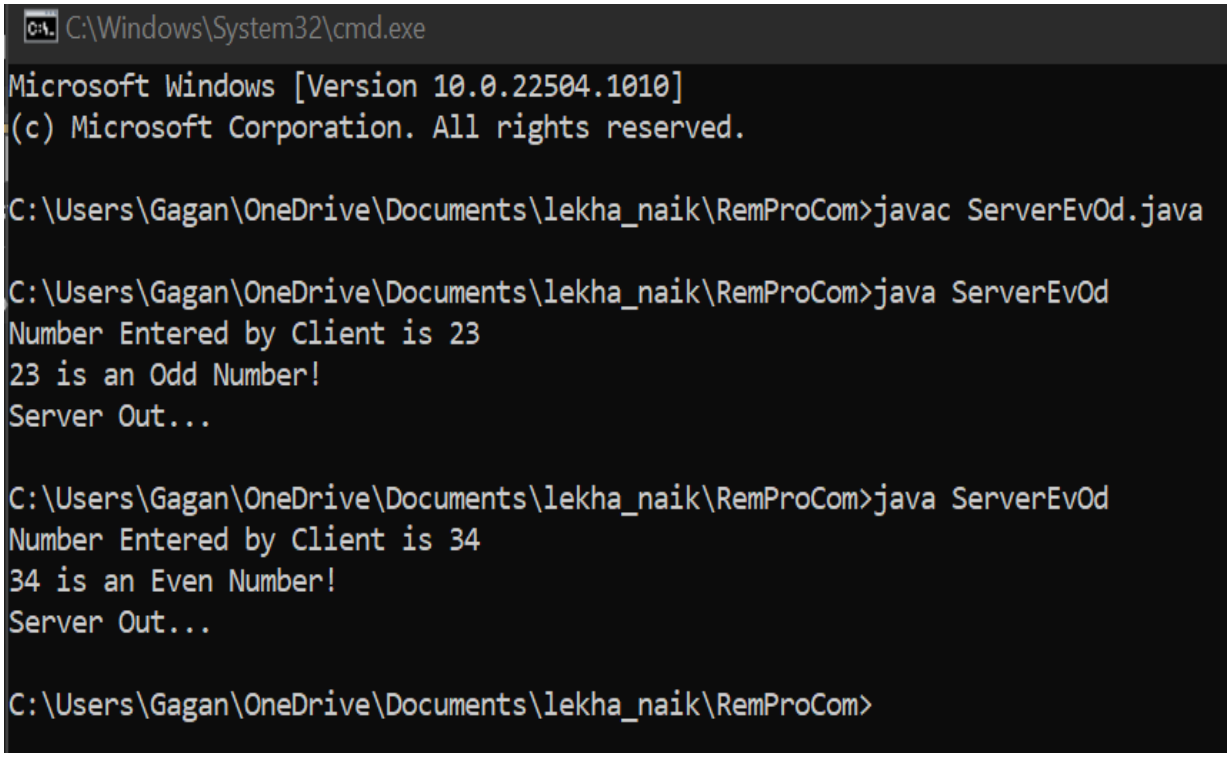
```
            System.out.println("Server Out...");
     }
}
```

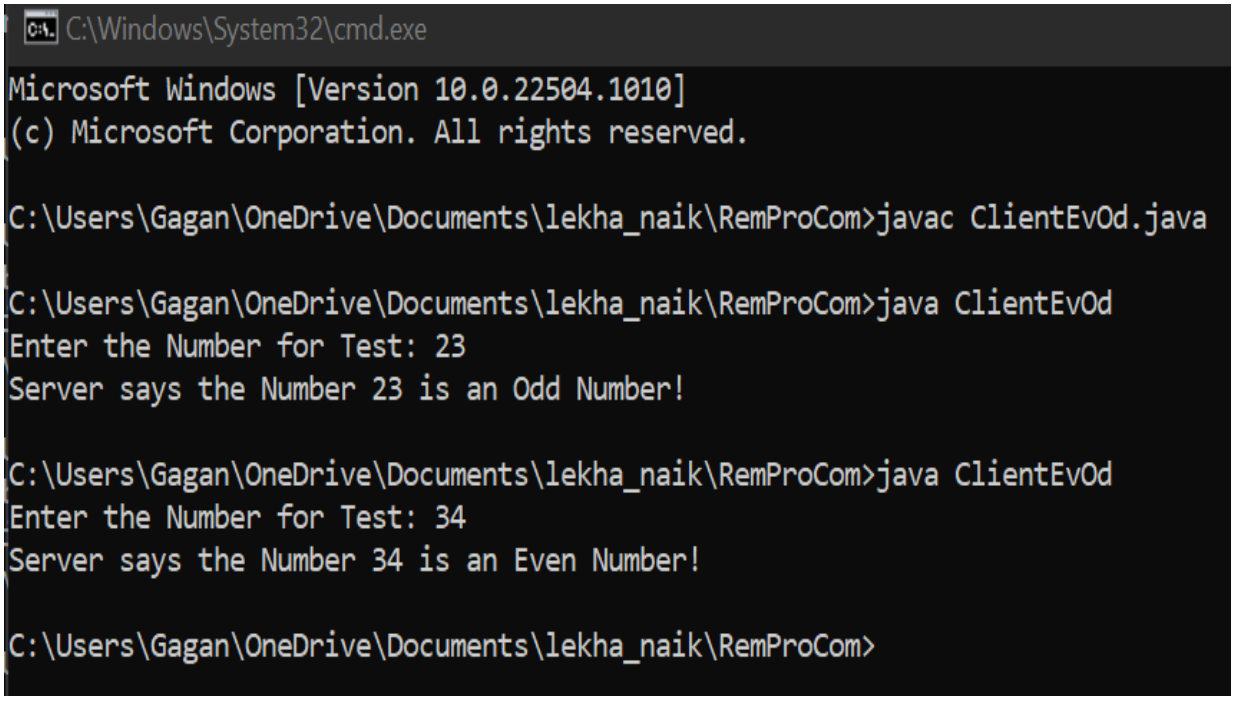
ClientEvOd.java

```
import java.io.IOException;
import java.net.*;
import java.util.*;
public class ClientEvOd
{
   public static void main(String args[]) throws IOException
   {
      Scanner sc = new Scanner(System.in);
      DatagramSocket ds = new DatagramSocket();
      InetAddress ip = InetAddress.getLocalHost();
      byte buf[] = null;
      DatagramPacket DpReceive = null;
      byte[] receive = new byte[65535];
      System.out.print("Enter the Number for Test: ");
         String inp = sc.nextLine();
         buf = inp.getBytes();
         DatagramPacket DpSend = new DatagramPacket(buf, buf.length, ip, 1234);
         ds.send(DpSend);
         DpReceive = new DatagramPacket(receive, receive.length);
         ds.receive(DpReceive);
         System.out.println("Server says the Number " + data(receive));
   }
   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:





Practical No. 01

Aim: 8) Implement a Program for multi-client chat server.

Code:

ServerMultiChat.java

```
import java.io.*;
import java.util.*;
import java.net.*;
public class ServerMultiChat
{
	static Vector<ClientThread> ar = new Vector<>();
	static int count = 0;
	public static void main(String[] args) throws IOException
	{
		ServerSocket srvsoc = new ServerSocket(1234);
		Socket soc;
		while (true)
		{
			soc = srvsoc.accept();
			System.out.println("New Client Request Received : " + soc);
			DataInputStream dis = new DataInputStream(soc.getInputStream());
			DataOutputStream dos = new DataOutputStream(soc.getOutputStream());
			System.out.println("Creating a New Handler for this Client...");
			ClientThread clt = new ClientThread(soc,"client " + count, dis, dos);
			Thread t = new Thread(clt);
			System.out.println("Adding this Client to Active Client List");
			ar.add(clt);
			t.start();
			count++;
		}
	}
}
class ClientThread implements Runnable
{
	Scanner snr = new Scanner(System.in);
	private String name;
	final DataInputStream dis;
	final DataOutputStream dos;
	Socket soc;
	boolean isloggedin;
	public ClientThread(Socket soc, String name, DataInputStream dis, DataOutputStream dos) {

		this.dis = dis;
```

```
                this.dos = dos;
                this.name = name;
                this.soc = soc;
                this.isloggedin=true;
        }
        public void run() {
                String receivedMsg;
                while (true)
                {
                        try
                        {
                                receivedMsg = dis.readUTF();
                                System.out.println(receivedMsg);
                                if(receivedMsg.equals("exit")){
                                        this.isloggedin=false;
                                        this.soc.close();
                                        break;
                                }
                                StringTokenizer st = new StringTokenizer(receivedMsg, "#");
                                String MsgToSend = st.nextToken();
                                String reciever = st.nextToken();
                                for (ClientThread ct : ServerMultiChat.ar)
                                {
                                        if (ct.name.equals(reciever) && ct.isloggedin==true)
                                        {
                                                ct.dos.writeUTF(this.name+" : "+MsgToSend);
                                                break;
                                        }
                                }
                        } catch (IOException e) {
                                e.printStackTrace();
                        }
                }
                try
                {
                        this.dis.close();
                        this.dos.close();
                }catch(IOException e){
                        e.printStackTrace();
                }
        }
}
```

ClientMultiChat.java

```
import java.net.*;

import java.io.*;

import java.util.*;

public class ClientMultiChat

{

	final static int port = 1234;

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

	{

		Socket soc  = new Socket ("localhost",port);

		Scanner in = new Scanner(System.in);

		DataInputStream dis = new DataInputStream(soc.getInputStream());

		DataOutputStream dos = new DataOutputStream(soc.getOutputStream());

		Thread sendMsg = new Thread(new Runnable()

		{

			public void run()

			{

				while(true)

				{

					String msg = in.nextLine();

					try

					{

						dos.writeUTF(msg);

					}

					catch(Exception e)
```

```
                                        {}
                                }
                        }
                });
Thread readMsg =new Thread(new Runnable()
{
        public void run()
        {
                while(true)
                {
                        try
                        {
                                String msg = dis.readUTF();
                                System.out.println(msg);
                        }
                        catch(Exception e)
                        {}
                }
        }
});
sendMsg.start();
readMsg.start();
}}
```

Output:

```
C:\Windows\System32\cmd.exe - java ServerMultiChat
Microsoft Windows [Version 10.0.22504.1010]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Gagan\OneDrive\Documents\lekha_naik\RemProCom>javac ServerMultiChat.java

C:\Users\Gagan\OneDrive\Documents\lekha_naik\RemProCom>java ServerMultiChat
New Client Request Received : Socket[addr=/127.0.0.1,port=52426,localport=1234]
Creating a New Handler for this Client...
Adding this Client to Active Client List
New Client Request Received : Socket[addr=/127.0.0.1,port=52427,localport=1234]
Creating a New Handler for this Client...
Adding this Client to Active Client List
Hi Lekha #client 1
Hi Naik #client 0
Which subject is this? #client 1
It is DS&CC Lab #client 0
```

```
C:\Windows\System32\cmd.exe - java ClientMultiChat
Microsoft Windows [Version 10.0.22504.1010]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Gagan\OneDrive\Documents\lekha_naik\RemProCom>java ClientMultiChat
client 0 : Hi Lekha
Hi Naik #client 0
client 0 : Which subject is this?
It is DS&CC Lab #client 0
```

```
C:\Windows\System32\cmd.exe - java ClientMultiChat
Microsoft Windows [Version 10.0.22504.1010]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Gagan\OneDrive\Documents\lekha_naik\RemProCom>javac ClientMultiChat.java

C:\Users\Gagan\OneDrive\Documents\lekha_naik\RemProCom>java ClientMultiChat
Hi Lekha #client 1
client 1 : Hi Naik
Which subject is this? #client 1
client 1 : It is DS&CC Lab
```

Practical No. 02

Programs on Remote Procedure Call

Aim: 1) To implement a Date Time Server using RPC concept. (Make use of datagram)

Code:

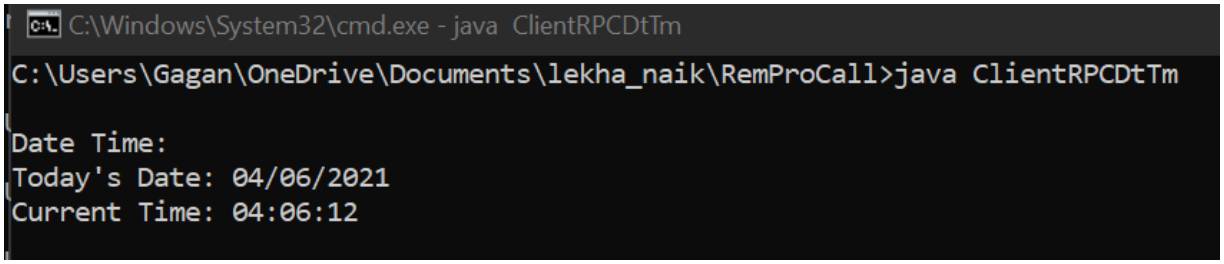
ServerRPCDtTm.java

```
import java.net.*;
import java.util.*;
import java.text.*;
class ServerRPCDtTm
{
	DatagramSocket ds;
	DatagramPacket dp;
	String result;
	ServerRPCDtTm()
	{
		try
		{
			ds = new DatagramSocket(2222);
			byte b[] = new byte[4096];
			dp = new DatagramPacket(b,b.length);
			ds.receive(dp);
			result= "Today's Date: "+fetchDate()+"\n"+"Current Time: "+fetchTime();
			InetAddress ia = InetAddress.getLocalHost();
			int port = dp.getPort();
			String dt = result;
			DatagramPacket op = new DatagramPacket(dt.getBytes(), dt.length(), ia,
port);
			ds.send(op);
		}
		catch (Exception e)
		{
			e.printStackTrace();
		}
	}
	public String fetchDate()
	{
		return new SimpleDateFormat("dd/mm/yyyy").format(new Date()).toString();
	}
	public String fetchTime()
	{
		return new SimpleDateFormat("hh:mm:ss").format(new Date()).toString();
	}
	public static void main(String[] args)
	{
		new ServerRPCDtTm();
	}
}
```

ClientRPCDtTm.java

```
import java.io.*;
import java.net.*;
class ClientRPCDtTm
{
	ClientRPCDtTm()
	{
		try
		{
			InetAddress ia = InetAddress.getLocalHost();
			DatagramSocket ds = new DatagramSocket(1300);
			while(true)
			{
			byte[] b = new byte[1024];
			DatagramPacket dp = new DatagramPacket(b,b.length,ia,2222);
			ds.send(dp);
			dp = new DatagramPacket(b,b.length);
			ds.receive(dp);
			String dt = new String(dp.getData(),0,dp.getLength());
			System.out.println("\nDate Time: \n"+dt+"\n");
			}
		}
		catch (Exception e)
		{
			e.printStackTrace();
		}
	}
	public static void main(String[] args)
	{
		new ClientRPCDtTm();
	}
}
```

Output:



Practical No. 02

Aim: 2) Write a program to implement a Server calculator containing ADD( ) , MUL( ) , SUB( ) , DIV( ). Implement using RPC.

Code:

ServerRPCOpr.java

```
import java.net.*;
import java.util.*;
class ServerRPCOpr
{
	DatagramSocket ds;
	DatagramPacket dp;
	String str, methodName, result;
	int val1, val2;
	ServerRPCOpr()
	{
		try
		{
			ds = new DatagramSocket(2222);
			byte b[] = new byte[4096];
			while(true)
			{
				dp = new DatagramPacket(b,b.length);
				ds.receive(dp);
				str = new String(dp.getData(),0,dp.getLength());
				if(str.equalsIgnoreCase("quit"))
				System.exit(1);
				else
				{
					StringTokenizer st = new StringTokenizer(str," ");
					while(st.hasMoreTokens()){
					String token = st.nextToken();
					methodName = token;
					val1 = Integer.parseInt(st.nextToken());
					val2 = Integer.parseInt(st.nextToken());
				}
			}
		System.out.println("\nClient Selected "+str+" Method : ");
		System.out.println("\nFirst Value : "+val1);
		System.out.println("Second Value :  "+val2);
		if(methodName.equals("Addition"))
		{
			result= "" + add(val1,val2);
		}
```

```
                if(methodName.equals("Subtraction"))
                {
                        result= "" + sub(val1,val2);
                }
                if(methodName.equals("Multiplication"))
                {
                        result= "" + mul(val1,val2);
                }
                if(methodName.equals("Division"))
                {
                        result= "" + div(val1,val2);
                }
                byte b1[] = result.getBytes();
                DatagramPacket dp1 = new
DatagramPacket(b1,b1.length,InetAddress.getLocalHost(), 1300);
                ds.send(dp1);
                }
        }
        catch (Exception e)
        {
                e.printStackTrace();
        }
}
public int add(int val1, int val2)
{
return val1 + val2;
}
public int sub(int val1, int val2)
{
return val1 - val2;
}
public int mul(int val1, int val2)
{
return val1 * val2;
}
public int div(int val1, int val2)
{
return val1 / val2;
}
public static void main(String[] args)
{
        new ServerRPCOpr();
}
}
```

ClientRPCOpr.java

```
import java.io.*;
import java.net.*;
class ClientRPCOpr
{
	ClientRPCOpr()
	{
		try		{
			InetAddress ia = InetAddress.getLocalHost();
			DatagramSocket ds = new DatagramSocket(1300);
			System.out.println("Enter Method Name with Parameter(e.g. [Addition 2 3]:
");			while(true)
			{
				BufferedReader br = new BufferedReader(new
InputStreamReader(System.in));
				String msg = br.readLine();
				DatagramPacket dp = new
DatagramPacket(msg.getBytes(),msg.length(),ia,2222);
				ds.send(dp);
				byte[] b = new byte[1024];
				dp = new DatagramPacket(b,b.length);
				ds.receive(dp);
				String s = new String(dp.getData(),0,dp.getLength());
				System.out.println("\nResult = " + s);
				System.out.println("\nEnter Method Name with Parameter(e.g.
[Addition 2 3]: ");
			}		}
		catch (Exception e)
		{
			e.printStackTrace();
		}		}
		public static void main(String[] args)
		{
			new ClientRPCOpr();
		}}
```

Output:

```
C:\Windows\System32\cmd.exe - java ServerRPCOpr
Microsoft Windows [Version 10.0.22504.1010]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Gagan\OneDrive\Documents\lekha_naik\RemProCall>javac ServerRPCOpr.java

C:\Users\Gagan\OneDrive\Documents\lekha_naik\RemProCall>java ServerRPCOpr

Client Selected Addition 23 34 Method :

First Value : 23
Second Value :  34

Client Selected Subtraction 34 23 Method :

First Value : 34
Second Value :  23

Client Selected Multiplication 2 3 Method :

First Value : 2
Second Value :  3

Client Selected Division 4 2 Method :

First Value : 4
Second Value :  2
```

```
C:\Windows\System32\cmd.exe - java ClientRPCOpr
Microsoft Windows [Version 10.0.22504.1010]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Gagan\OneDrive\Documents\lekha_naik\RemProCall>javac ClientRPCOpr.java

C:\Users\Gagan\OneDrive\Documents\lekha_naik\RemProCall>java ClientRPCOpr
Enter Method Name with Parameter(e.g. [Addition 2 3]:
Addition 23 34

Result = 57

Enter Method Name with Parameter(e.g. [Addition 2 3]:
Subtraction 34 23

Result = 11

Enter Method Name with Parameter(e.g. [Addition 2 3]:
Multiplication 2 3

Result = 6

Enter Method Name with Parameter(e.g. [Addition 2 3]:
Division 4 2

Result = 2
```

Practical No. 02

Aim: 3) Implement a Server to do the following: (Use RPC)

i) Get two numbers from the client.
ii) Server processing the summation of the above two numbers
iii) Server sends the processed data to the client and client checks whether the sum is greater than 100 or not
iv) Client displays appropriate message

Code:

ServerAddGrt.java

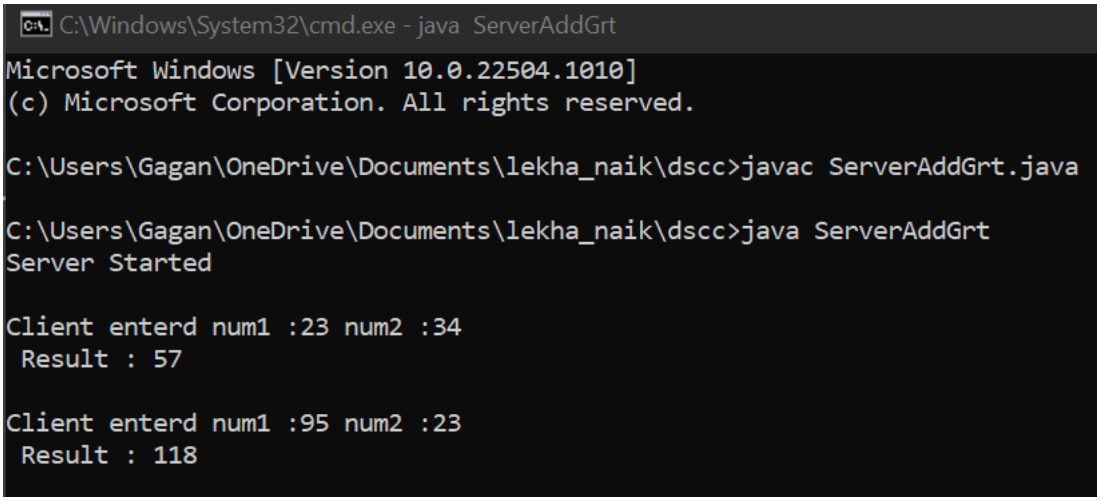
```
import java.net.*;
import java.io.*;
public class ServerAddGrt
{
	private static final int PORT=2222;
	public static void main(String[] args)throws Exception
	{
		ServerSocket srvsoc=new ServerSocket(PORT);


		System.out.println("Server Started");
		Socket soc = srvsoc.accept();while(true)
		{
		DataInputStream dis=new DataInputStream(soc.getInputStream());
		DataOutputStream dos=new DataOutputStream(soc.getOutputStream());
		int num1 = dis.read();
		int num2= dis.read();
		int add = num1 + num2;
		System.out.println("\nClient enterd num1 :"+num1+" num2 :"+num2+"\n Result :
"+add);
		dos.write(add);
		}
	}
}
```

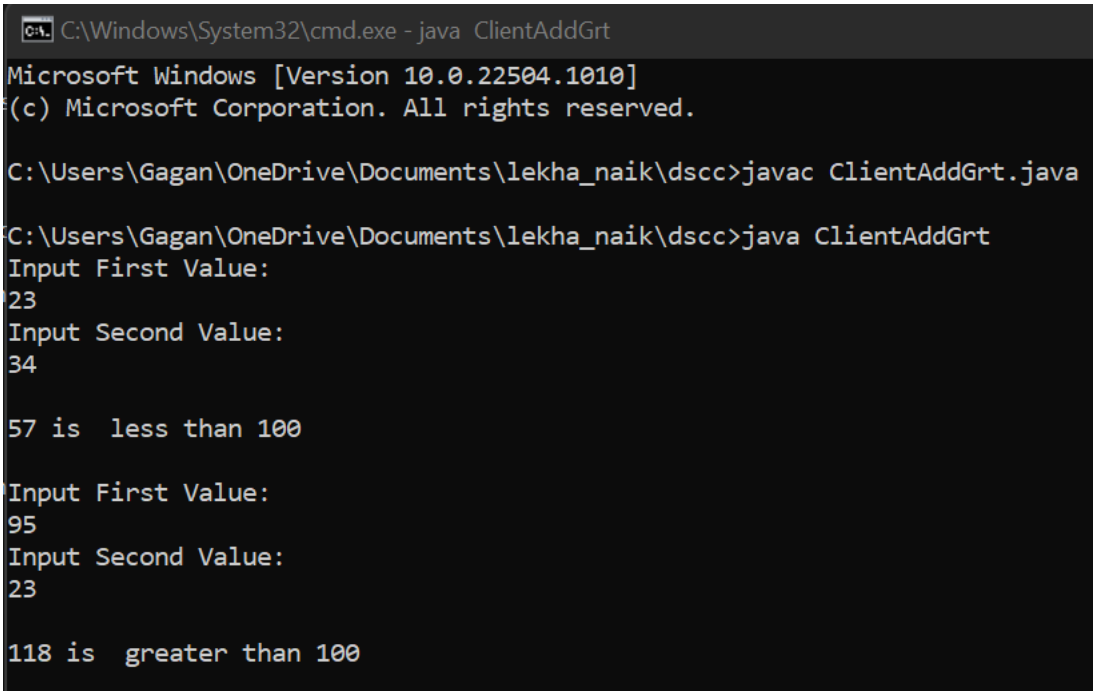
ClientAddGrt.java

```
import java.net.*;
import java.io.*;
import java.util.*;
public class ClientAddGrt
{
	private static InetAddress host;
	private static final int PORT=2222;
	public static void main(String[] args)throws Exception
	{
		try
		{
			host=InetAddress.getLocalHost();
		}
		catch(UnknownHostException e)
		{
			System.out.println("Host Not Found");
			System.exit(1);
		}
		int num1,num2;
		Socket soc=new Socket(host,PORT);
		DataInputStream dis=new DataInputStream(soc.getInputStream());
		DataOutputStream dos=new DataOutputStream(soc.getOutputStream());
		Scanner input=new Scanner(System.in);
		while(true)
		{
			System.out.println("Input First Value: ");
			num1=input.nextInt();
			System.out.println("Input Second Value: ");
			num2=input.nextInt();
			dos.write(num1);
```

```
                    dos.write(num2);
                    int add=dis.read();
                    if(add>100) {
                            System.out.println("\n"+add+" is  greater than 100\n");}
                    else
                    {

                            System.out.println("\n"+add+" is  less than 100\n");
                    }               }       }
}
```

Output:





Practical No. 03

Programs on Remote Method Invocation

Aim: 1) Retrieve time and date function from server to client. This program should display server date and time by implementing RMI.

Code:

RMIDtTmIntrfc.java

```
import java.rmi.*;

public interface RMIDtTmIntrfc extends Remote

{

        public String getDate()throws RemoteException;

        public String getTime() throws RemoteException;

}
```

RMIDtTmImpl.java

```
import java.rmi.*;

import java.rmi.server.*;

import java.util.*;

import java.text.*;

public class RMIDtTmImpl extends UnicastRemoteObject implements RMIDtTmIntrfc

{

        RMIDtTmImpl() throws RemoteException

        {

                super();

        }

        public String getDate()throws RemoteException

        {

                return new SimpleDateFormat("dd/mm/yyyy").format(new Date()).toString();

        }

        public String getTime()throws RemoteException

        {

                return new SimpleDateFormat("hh:mm:ss").format(new Date()).toString();

        }}
```

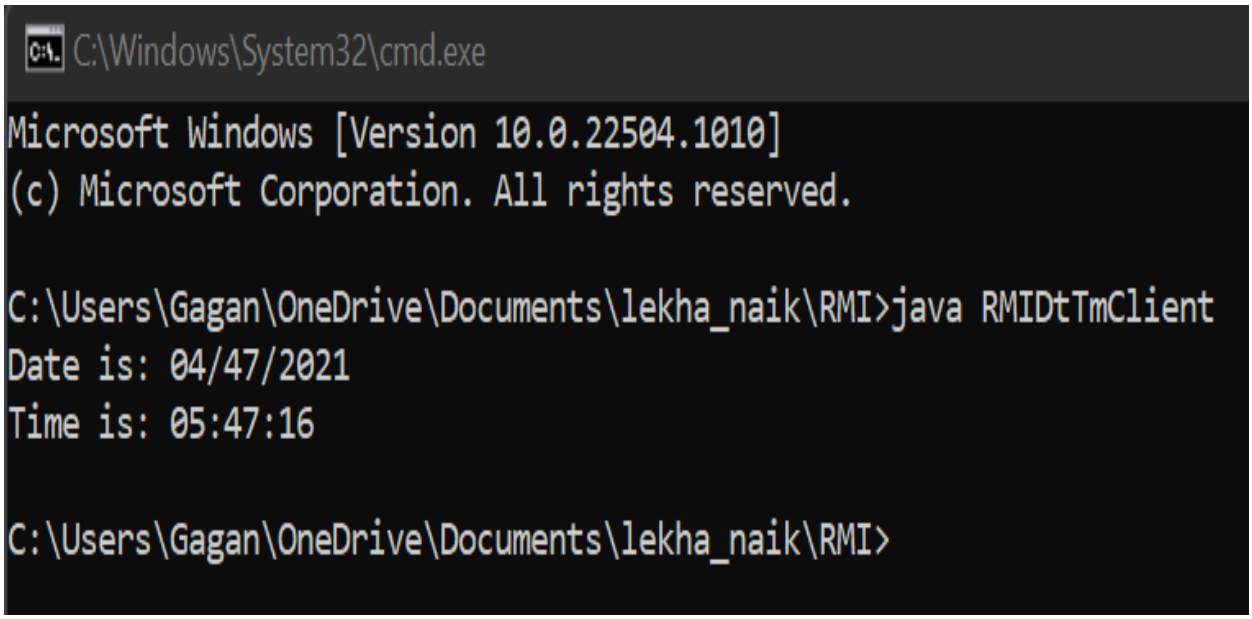
RMIDtTmRegister.java

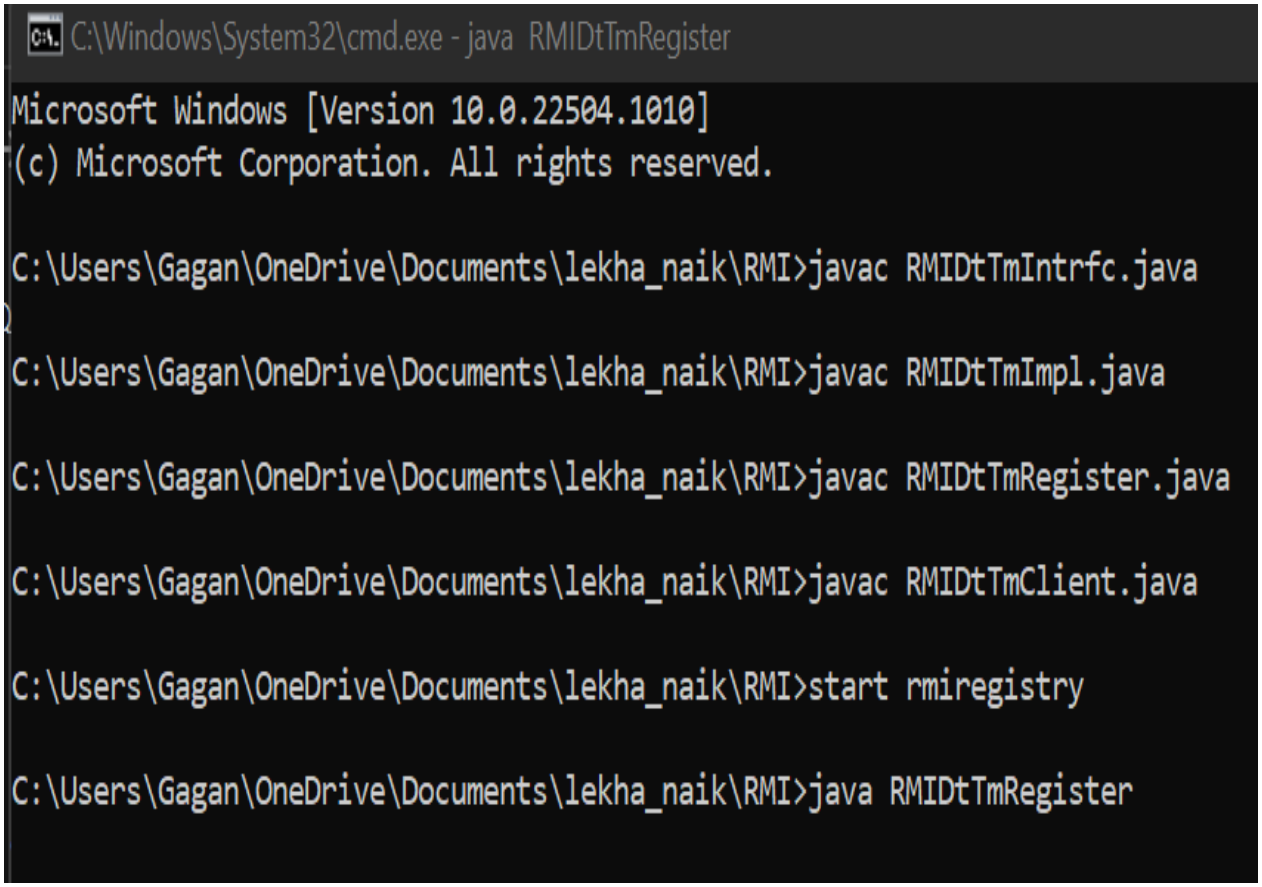
```
import java.rmi.*;

import java.rmi.registry.*;

public class RMIDtTmRegister

{

	public static void main(String[] args)

	{

		try

		{

			Registry reg = LocateRegistry.getRegistry();

			RMIDtTmImpl obj = new RMIDtTmImpl();

			Naming.rebind("RMIDtTmImpl",obj);

		}

		catch(Exception e)

		{			}

	}	}
```

RMIDtTmClient.java

```
import java.rmi.*;

public class RMIDtTmClient

{

	public static void main(String[] args)

	{

		try

		{

			RMIDtTmIntrfc obj=(RMIDtTmIntrfc)Naming.lookup("RMIDtTmImpl");

			System.out.println("Date is: "+obj.getDate());

			System.out.println("Time is: "+obj.getTime());

		}

		catch(Exception e)

		{			}

	}

}
```

Output:


C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22504.1010]
(c) Microsoft Corporation. All rights reserved.
C:\Users\Gagan\OneDrive\Documents\lekha_naik\RMI>java RMIDtTmClient
Date is: 04/47/2021
Time is: 05:47:16
C:\Users\Gagan\OneDrive\Documents\lekha_naik\RMI>


C:\Windows\System32\cmd.exe - java RMIDtTmRegister
Microsoft Windows [Version 10.0.22504.1010]
(c) Microsoft Corporation. All rights reserved.
C:\Users\Gagan\OneDrive\Documents\lekha_naik\RMI>javac RMIDtTmIntrfc.java
C:\Users\Gagan\OneDrive\Documents\lekha_naik\RMI>javac RMIDtTmImpl.java
C:\Users\Gagan\OneDrive\Documents\lekha_naik\RMI>javac RMIDtTmRegister.java
C:\Users\Gagan\OneDrive\Documents\lekha_naik\RMI>javac RMIDtTmClient.java
C:\Users\Gagan\OneDrive\Documents\lekha_naik\RMI>start rmiregistry
C:\Users\Gagan\OneDrive\Documents\lekha_naik\RMI>java RMIDtTmRegister

Practical No. 03

Programs on Remote Method Invocation

Aim: 2) The client should provide the values of a, b. The server will solve the equation c = (a+b)2 & c= (a+b)3 and will give back the value of c. Implement using RMI.

Code:

IntrfcEq.java

```
import java.rmi.*;

public interface IntrfcEq extends Remote

{

        public double square(int a, int b) throws RemoteException;

        public double cube(int a, int b) throws RemoteException;

}
```

IntrfcEq.java

```
import java.rmi.*;

import java.rmi.server.*;

public class ImplEq extends UnicastRemoteObject implements IntrfcEq

{

        public ImplEq()throws RemoteException

        {

                super();

        }

        public double square(int a, int b)throws RemoteException

        {

                double ans = (a*a)+(b*b)+2*a*b;

                return ans;

        }

        public double cube(int a, int b)throws RemoteException

        {

                double ans = (a*a*a)+3*a*a*b+3*a*(b*b)+(b*b*b);

                return ans;

        }}
```

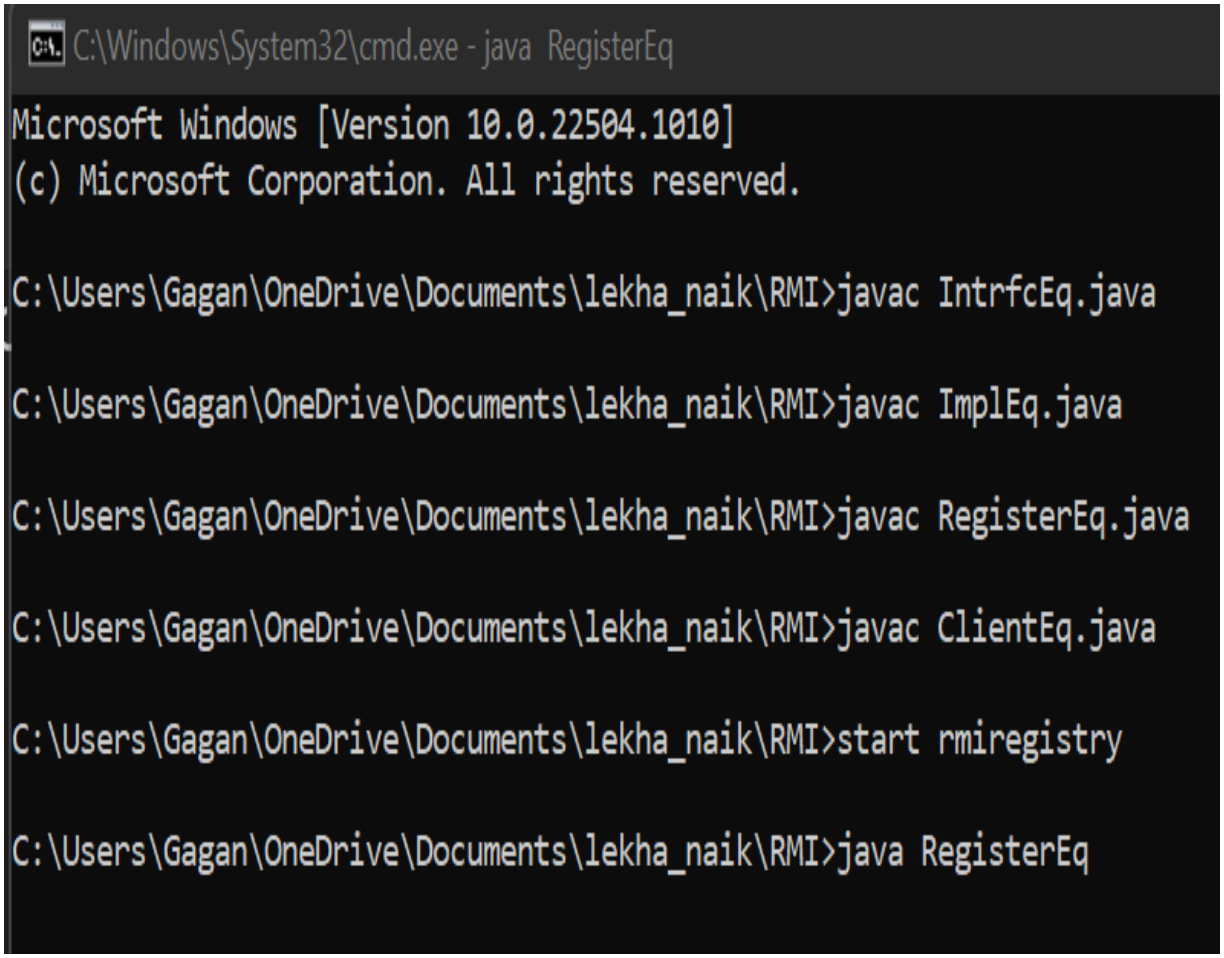
RegisterEq.java

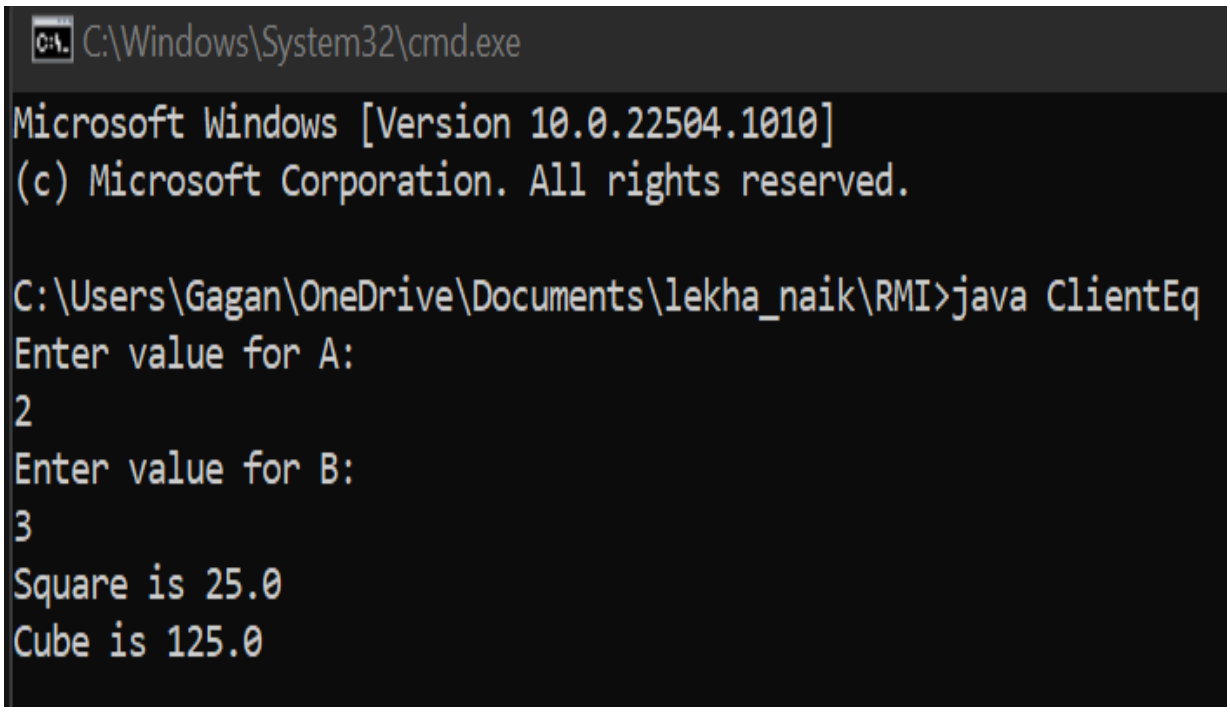
```
import java.rmi.*;

import java.rmi.registry.*;

public class RegisterEq

{	public static void main(String[] args)

	{

		try		{

			Registry reg = LocateRegistry.getRegistry();

			ImplEq obj = new ImplEq();

			Naming.rebind("ImplEq",obj);

		}

		catch(Exception e)

		{		}

	}}
```

ClientEq.java

```
import java.rmi.*;

import java.util.*;

public class ClientEq

{

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

	{

		int a, b;

		double ans;

		Scanner input = new Scanner(System.in);

		IntrfcEq obj = (IntrfcEq)Naming.lookup("ImplEq");

		System.out.println("Enter value for A: ");

		a=input.nextInt();

		System.out.println("Enter value for B: ");

		b=input.nextInt();

		System.out.println("Square is "+obj.square(a,b));

		System.out.println("Cube is "+obj.cube(a,b));

	}}
```

Output:


C:\Windows\System32\cmd.exe - java RegisterEq
Microsoft Windows [Version 10.0.22504.1010]
(c) Microsoft Corporation. All rights reserved.
C:\Users\Gagan\OneDrive\Documents\lekha_naik\RMI>javac IntrfcEq.java
C:\Users\Gagan\OneDrive\Documents\lekha_naik\RMI>javac ImplEq.java
C:\Users\Gagan\OneDrive\Documents\lekha_naik\RMI>javac RegisterEq.java
C:\Users\Gagan\OneDrive\Documents\lekha_naik\RMI>javac ClientEq.java
C:\Users\Gagan\OneDrive\Documents\lekha_naik\RMI>start rmiregistry
C:\Users\Gagan\OneDrive\Documents\lekha_naik\RMI>java RegisterEq


C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22504.1010]
(c) Microsoft Corporation. All rights reserved.
C:\Users\Gagan\OneDrive\Documents\lekha_naik\RMI>java ClientEq
Enter value for A:
2
Enter value for B:
3
Square is 25.0
Cube is 125.0

Practical No. 03

Aim: 3) The client should provide the values of a, b & c. The server will solve the equation (ax2 + bx + c = 0) and will give back the value of x. If a = 1, b = 5 and c = 6 then return value will be x = -2 or x = -3.

Code:

IntrfcEqQud.java

```
import java.rmi.*;

public interface IntrfcEqQud extends Remote

{

	public double quad(int a, int b, int c) throws RemoteException;

}
```

ImplEqQud.java

```
import java.rmi.*;

import java.rmi.server.*;

public class ImplEqQud extends UnicastRemoteObject implements IntrfcEqQud

{

	public ImplEqQud()throws RemoteException

	{

		super();

	}

	public double quad(int a, int b, int c)throws RemoteException

	{

		double x = (b*b)-(4*a*c);

		double ans = (float)(-b-Math.sqrt(x))/(2*a);

		return ans;

	}

}
```

RegisterEqQud.java

```
import java.rmi.*;

import java.rmi.registry.*;

public class RegisterEqQud

{
```

```
	public static void main(String[] args)
	{
		try		{
			Registry reg = LocateRegistry.getRegistry();
			ImplEqQud obj = new ImplEqQud();
			Naming.rebind("ImplEqQud",obj);
		}
		catch(Exception e)
		{		}
	}
}
```

ClientcEqQud.java

```
import java.rmi.*;
import java.util.*;
public class ClientEqQud
{
	public static void main(String[] args)throws Exception
	{
		int a, b, c;
		double ans;
		Scanner input = new Scanner(System.in);
		IntrfcEqQud obj = (IntrfcEqQud)Naming.lookup("ImplEqQud");
		System.out.println("Enter value for A: ");
		a=input.nextInt();
		System.out.println("Enter value for B: ");
		b=input.nextInt();
		System.out.println("Enter value for C: ");
		c=input.nextInt();
		System.out.println("\nResult Factor: "+obj.quad(a,b,c));
	}
}
```

Output:

```
C:\Windows\System32\cmd.exe - java RegisterEqQud
Microsoft Windows [Version 10.0.22504.1010]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Gagan\OneDrive\Documents\lekha_naik\RMI>javac IntrfcEqQud.java

C:\Users\Gagan\OneDrive\Documents\lekha_naik\RMI>javac ImplEqQud.java

C:\Users\Gagan\OneDrive\Documents\lekha_naik\RMI>javac RegisterEqQud.java

C:\Users\Gagan\OneDrive\Documents\lekha_naik\RMI>javac ClientEqQud.java

C:\Users\Gagan\OneDrive\Documents\lekha_naik\RMI>start rmiregistry

C:\Users\Gagan\OneDrive\Documents\lekha_naik\RMI>java RegisterEqQud
```

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22504.1010]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Gagan\OneDrive\Documents\lekha_naik\RMI>java ClientEqQud
Enter value for A:
1
Enter value for B:
5
Enter value for C:
6

Result Factor: -3.0
```

Practical No. 03

Aim: 4) Design a Graphical User Interface to find Student's score. Implement using RMI.

Code:

IntrfcStu.java

```
import java.rmi.*;

public interface IntrfcStu extends Remote

{

        public double findScore(String name)throws RemoteException;

}
```

ImplStu.java

```
import java.rmi.*;

import java.rmi.server.*;

import java.util.*;

 public class ImplStu extends UnicastRemoteObject implements IntrfcStu

 {

        private HashMap<String,Double> scores = new HashMap<String,Double>();

        public ImplStu()throws RemoteException

        {

                super();

                initializeStudent();

        }

        public void initializeStudent()

        {

                scores.put("Rachel",new Double(75.00));

                scores.put("Monica",new Double(95.00));

                scores.put("Ross",new Double(85.00));

                scores.put("Chandler",new Double(80.00));

        }

 public double findScore(String name)throws RemoteException

 {
```

```
        Double d = (Double)scores.get(name);
        if(d==null)
        {
                System.out.println("Student"+name+"is not found");
                return -1;
        }
        else
        {
                System.out.println("Student "+name+" score is "+d.doubleValue());
                return d;
        }
}
}
```

RegisterStu.java

```
import java.rmi.*;
import java.rmi.registry.*;
public class RegisterStu
{
        public static void main(String[] agrs)
        {
                try
                {
                        ImplStu obj = new ImplStu();
                        Registry reg = LocateRegistry.getRegistry();
                        Naming.rebind("ImplStu",obj);
                }
                 catch(Exception e)
                 {}
        }
}
```

ClientStu.java

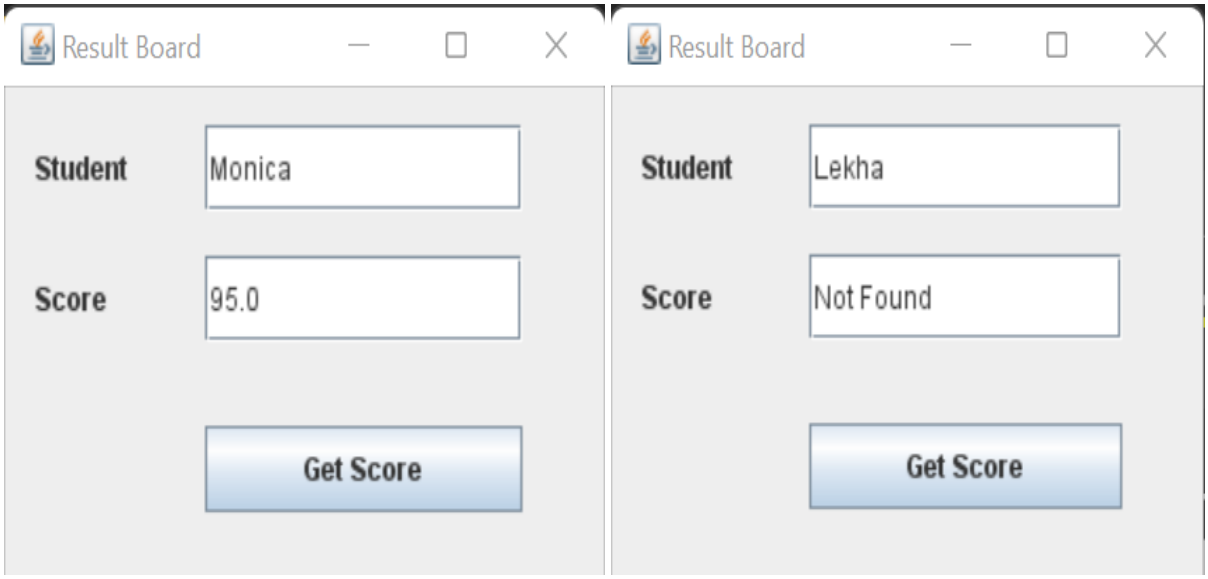
```
import java.rmi.*;
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
public class ClientStu extends JApplet
{
	private IntrfcStu student;
	private JButton getScore = new JButton("Get Score");
	private JTextField name = new JTextField();
	private JTextField tfscore = new JTextField();
	private JLabel lname = new JLabel("Student");
	private JLabel lscore = new JLabel("Score");
	public void init()
	{
		try
		{
			student = (IntrfcStu)Naming.lookup("ImplStu");
		}
		catch(Exception e)
		{}
		JPanel panel = new JPanel();
		panel.setLayout(null);
		lname.setBounds(15, 15, 75, 33);
		panel.add(lname);
		name.setBounds(95, 15, 150, 33);
		panel.add(name);
		lscore.setBounds(15, 65, 75, 33);
		panel.add(lscore);
		tfscore.setBounds(95, 65, 150, 33);
		panel.add(tfscore);
```

```
                getScore.setBounds(95, 130, 150, 33);
                panel.add(getScore, BorderLayout.CENTER);
                add(panel,BorderLayout.CENTER);
                getScore.addActionListener(new ActionListener()
                {
                        public void actionPerformed(ActionEvent ae)
                        {
                                getScore();
                        }               });
}
public void getScore()
{
        try     {
                double score = student.findScore(name.getText());
                if(score<0)
                        tfscore.setText("Not Found");
                else
                tfscore.setText(""+score);
        }
        catch(Exception ex)
        {       }}
public static void main(String[] args)
{
        ClientStu applet = new ClientStu();
        JFrame frame = new JFrame();
        frame.setTitle("Result Board");
        frame.add(applet,BorderLayout.CENTER);
        frame.setSize(300,250);
        applet.init();
        frame.setVisible(true);
}}
```

Output:

```
C:\Windows\System32\cmd.exe - java RegisterStu
Microsoft Windows [Version 10.0.22504.1010]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Gagan\OneDrive\Documents\lekha_naik\RMI>javac IntrfcStu.java

C:\Users\Gagan\OneDrive\Documents\lekha_naik\RMI>javac ImplStu.java

C:\Users\Gagan\OneDrive\Documents\lekha_naik\RMI>javac RegisterStu.java

C:\Users\Gagan\OneDrive\Documents\lekha_naik\RMI>javac ClientStu.java

C:\Users\Gagan\OneDrive\Documents\lekha_naik\RMI>start rmiregistry

C:\Users\Gagan\OneDrive\Documents\lekha_naik\RMI>java RegisterStu
Student Rachel score is 75.0
Student Monica score is 95.0
StudentLekhais not found
```

```
C:\Windows\System32\cmd.exe - java ClientStu
Microsoft Windows [Version 10.0.22504.1010]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Gagan\OneDrive\Documents\lekha_naik\RMI>java ClientStu
```



Practical No. 03

Aim: 5) Design a Graphical User Interface to find greatest of two numbers. Implement using RMI.

Code:

IntrfcGrt.java

```
import java.rmi.*;


public interface IntrfcGrt extends Remote
{
	public boolean grt(int num1, int num2)throws RemoteException;
}
```

ImplcGrt.java

```
import java.rmi.*;
import java.rmi.server.*;


public class ImplGrt extends UnicastRemoteObject implements IntrfcGrt
{
	ImplGrt()throws RemoteException
	{
		super();
	}
	public boolean grt(int num1, int num2)throws RemoteException
	{
		boolean ans;
		if(num1>num2)
			return true;
		else
			return false;
	}
}
```

RegisterGrt.java

```
import java.rmi.*;
import java.rmi.registry.*;

public class RegisterGrt
{
	public static void main(String[] args)
	{
		try
		{
			Registry reg=LocateRegistry.createRegistry(3333);
			ImplGrt obj=new ImplGrt();
			Naming.rebind("ImplGrt",obj);
		}
		catch(Exception e)
		{
		}
	}
}
```

ClientGrt.java

```
import java.rmi.*;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class ClientGrt extends JFrame implements ActionListener
{
	JTextField tf1,tf2;
	JLabel lb1,lb2,lb3,lb4,lb5;
	JButton btn;
```

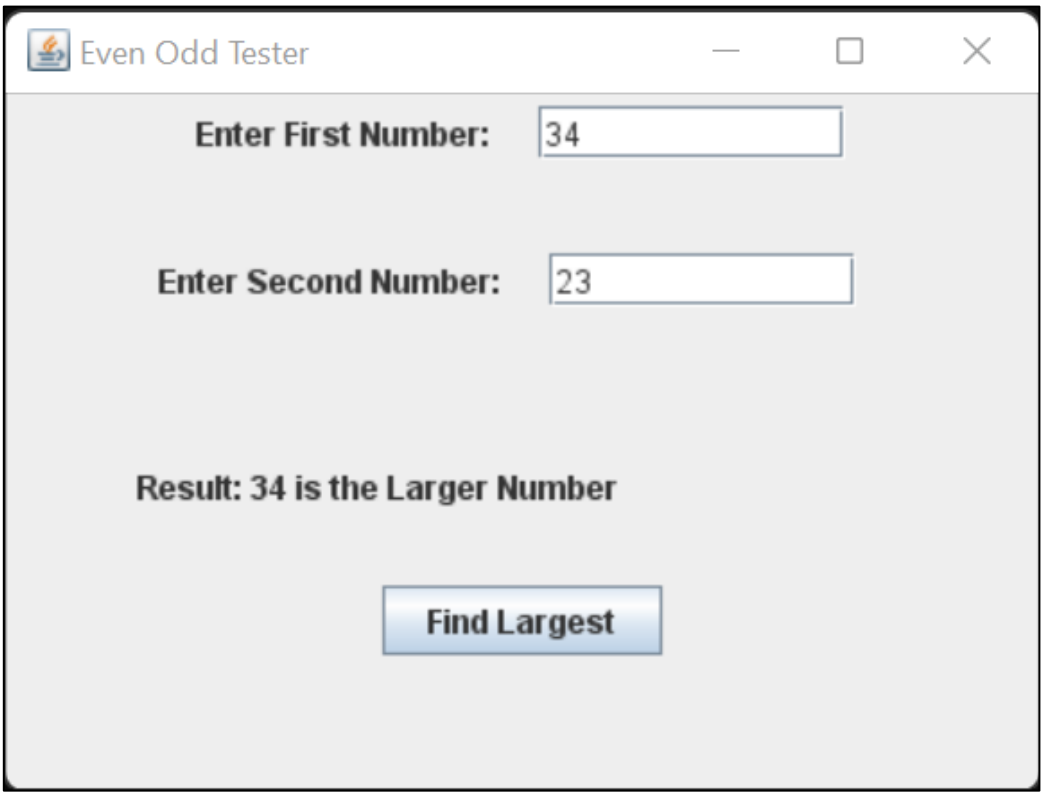
```
        ClientGrt()
        {
                JFrame frame = new JFrame("Even Odd Tester");
                tf1=new JTextField(10);
                tf2=new JTextField(10);
                lb1=new JLabel("<html>Enter First Number: </html>",
SwingConstants.CENTER);
                lb5=new JLabel("<html>Enter Second Number: </html>",
SwingConstants.CENTER);
                lb2=new JLabel("<html><br/><br/><br/><br/><br/></html>",
SwingConstants.CENTER);
                lb3=new JLabel("", SwingConstants.CENTER);
                lb4=new JLabel("<html><br/><br/><br/><br/></html>", SwingConstants.CENTER);
                btn=new JButton("Find Largest");
    Panel p=new Panel();
        p.add(lb1);
                p.add(tf1);
                p.add(lb5);
                p.add(tf2);
                p.add(lb2);
                p.add(lb3);
                p.add(lb4);
                p.add(btn);
    frame.add(p);
                frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(400, 300);
        frame.setLocationRelativeTo(null);
        frame.setVisible(true);
                btn.addActionListener(this);
        }

        public void actionPerformed(ActionEvent ae)
```

```
	{
		try
		{
			IntrfcGrt obj=(IntrfcGrt)Naming.lookup("ImplGrt");
			int num1=Integer.parseInt(tf1.getText());
			int num2=Integer.parseInt(tf2.getText());
			if(obj.grt(num1,num2)){
				lb3.setText("                    Result: "+num1+" is the
Larger Number                    ");
			}
			else
			{
				lb3.setText("                Result: "+num2+" is the Larger
Number                                    ");
			}
		}
		catch(Exception e)
		{
			lb3.setText("Error");
		}
	}

	public static void main(String[] args)
	{
		ClientGrt gc=new ClientGrt();
		gc.setLayout(new GridLayout(6,1));
		gc.setVisible(true);
		gc.setSize(400,300);
	}
}
```

Output:

```
C:\Windows\System32\cmd.exe - java  RegisterGrt
Microsoft Windows [Version 10.0.22504.1010]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Gagan\OneDrive\Documents\lekha_naik\RMI>javac IntrfcGrt.java

C:\Users\Gagan\OneDrive\Documents\lekha_naik\RMI>javac ImplGrt.java

C:\Users\Gagan\OneDrive\Documents\lekha_naik\RMI>javac RegisterGrt.java

C:\Users\Gagan\OneDrive\Documents\lekha_naik\RMI>javac ClientGrt.java

C:\Users\Gagan\OneDrive\Documents\lekha_naik\RMI>start rmiregistry

C:\Users\Gagan\OneDrive\Documents\lekha_naik\RMI>java RegisterGrt
```

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22504.1010]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Gagan\OneDrive\Documents\lekha_naik\RMI>java ClientGrt
```



Practical No. 03

Aim: 6) Design a Graphical User Interface (GUI) based Basic calculator by implementing RMI.

Code:

CalcIntrfc.java

```
import java.rmi.*;

public interface CalcIntrfc extends Remote

{

        public double add(double num1, double num2)throws RemoteException;

        public double sub(double num1, double num2)throws RemoteException;

        public double mul(double num1, double num2)throws RemoteException;

        public double div(double num1, double num2)throws RemoteException;

}
```

CalcImpl.java

```
import java.rmi.*;

import java.rmi.server.*;

import java.util.*;

 public class CalcImpl extends UnicastRemoteObject implements CalcIntrfc

 {

         public CalcImpl()throws RemoteException

         {

                  super();

         }

         public double add(double a, double b)throws RemoteException

         {

                 return a+b;

         }

         public double sub(double a, double b)throws RemoteException

         {

                 return a-b;

         }
```

```
        public double mul(double a, double b)throws RemoteException
        {
                return a*b;
        }
        public double div(double a, double b)throws RemoteException
        {
                return a/b;
        }
}
```

CalcRegister.java

```
import java.rmi.*;
import java.rmi.registry.*;
public class CalcRegister
{
        public static void main(String[] agrs)
        {
                try
                {
                        CalcImpl obj = new CalcImpl();
                        Registry reg = LocateRegistry.getRegistry();
                        Naming.rebind("CalcImpl",obj);
                }
                 catch(Exception e)
                 {}
        }
}
```

CalcClient.java

```
import java.rmi.*;
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
public class CalcClient extends JApplet
```

```
{
        private CalcIntrfc student;
        private JButton add = new JButton("+");
        private JButton sub = new JButton("-");
        private JButton mul = new JButton("*");
        private JButton div = new JButton("/");
        private JTextField tf1 = new JTextField();
        private JTextField tf2 = new JTextField();
        private JLabel lname = new JLabel("Number 1");
        private JLabel lscore = new JLabel("Number 2");
        private JLabel result = new JLabel("");
        public void init()
        {
                try
                {
                        student = (CalcIntrfc)Naming.lookup("CalcImpl");
                }
                catch(Exception e)
                {}
                JPanel panel = new JPanel();
                panel.setLayout(null);
                lname.setBounds(15, 15, 75, 33);
                panel.add(lname);
                tf1.setBounds(95, 15, 150, 33);
                panel.add(tf1);
                lscore.setBounds(15, 65, 75, 33);
                panel.add(lscore);
                tf2.setBounds(95, 65, 150, 33);
                panel.add(tf2);
                add.setBounds(95, 130, 100, 33);
                panel.add(add, BorderLayout.CENTER);
```

```
sub.setBounds(205, 130, 100, 33);
panel.add(sub, BorderLayout.CENTER);
mul.setBounds(315, 130, 100, 33);
panel.add(mul, BorderLayout.CENTER);
div.setBounds(425, 130, 100, 33);
panel.add(div, BorderLayout.CENTER);
result.setBounds(100, 170, 175, 75);
panel.add(result);
add(panel,BorderLayout.CENTER);
add.addActionListener(new ActionListener()
{
        public void actionPerformed(ActionEvent ae)
        {
                add();
        }
});
sub.addActionListener(new ActionListener()
{
        public void actionPerformed(ActionEvent ae)
        {
                sub();
        }
});
mul.addActionListener(new ActionListener()
{
        public void actionPerformed(ActionEvent ae)
        {
                mul();
        }
});
div.addActionListener(new ActionListener()
```

```
                {
                        public void actionPerformed(ActionEvent ae)
                        {
                                div();
                        }
                });
}
public void add()
{
        try
        {
                double score =
student.add(Double.parseDouble(tf1.getText()),Double.parseDouble(tf2.getText()));
                result.setText("Result: "+score);
                System.out.println("Result: "+score);
        }
        catch(Exception ex)
        {
        }
}
public void sub()
{
        try
        {
                double score =
student.sub(Double.parseDouble(tf1.getText()),Double.parseDouble(tf2.getText()));
                result.setText("Result: "+score);
                System.out.println("Result: "+score);
        }
        catch(Exception ex)
        {
        }
```

```
}
public void mul()
{
	try
	{
		double score =
student.mul(Double.parseDouble(tf1.getText()),Double.parseDouble(tf2.getText()));
		result.setText("Result: "+score);
		System.out.println("Result: "+score);
	}
	catch(Exception ex)
	{
	}
}
public void div()
{
	try
	{
		double score =
student.div(Double.parseDouble(tf1.getText()),Double.parseDouble(tf2.getText()));
		result.setText("Result: "+score);
		System.out.println("Result: "+score);
	}
	catch(Exception ex)
	{
	}
}
public static void main(String[] args)
{
	CalcClient applet = new CalcClient();
	JFrame frame = new JFrame();
	frame.setTitle("Calculator");
```

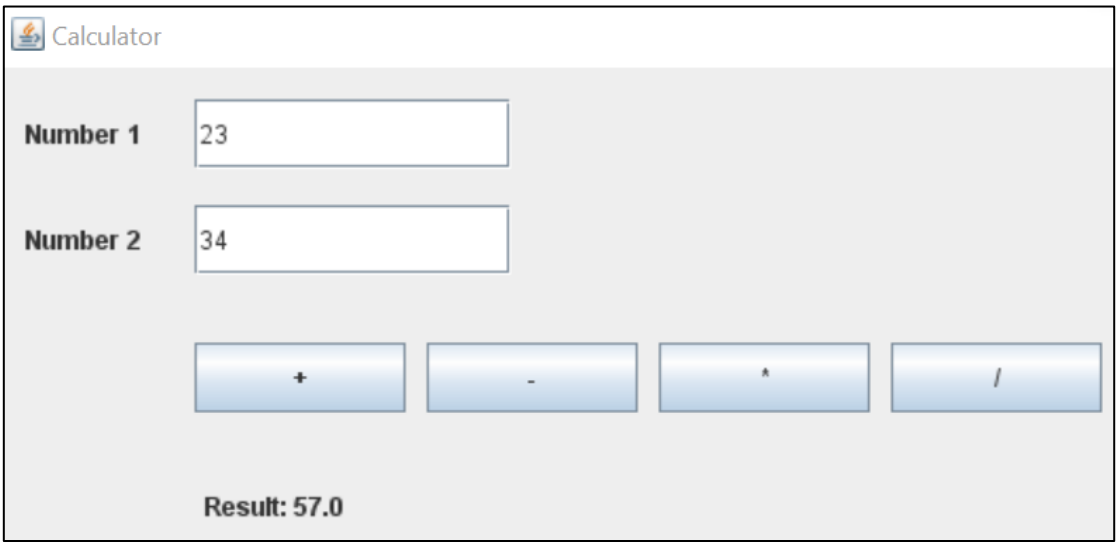
```
        frame.add(applet,BorderLayout.CENTER);
        frame.setSize(700,350);
        applet.init();
        frame.setVisible(true);
}}
```

Output:

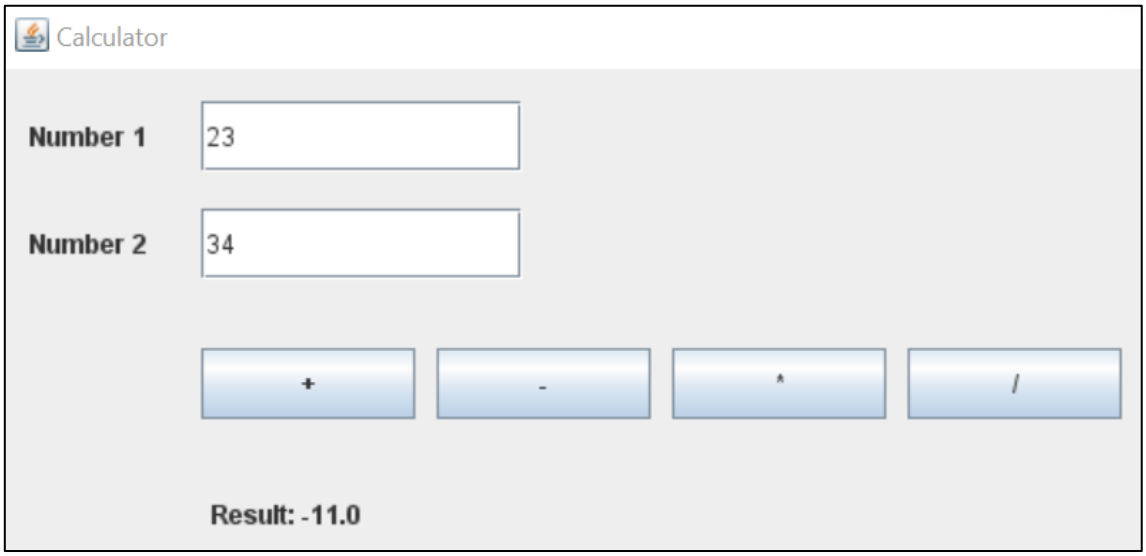
```
C:\Windows\System32\cmd.exe - java CalcRegister
Microsoft Windows [Version 10.0.22504.1010]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Gagan\OneDrive\Documents\lekha_naik\RMI\CalcRMI>javac CalcIntrfc.java

C:\Users\Gagan\OneDrive\Documents\lekha_naik\RMI\CalcRMI>javac CalcImpl.java

C:\Users\Gagan\OneDrive\Documents\lekha_naik\RMI\CalcRMI>javac CalcRegister.java

C:\Users\Gagan\OneDrive\Documents\lekha_naik\RMI\CalcRMI>javac CalcClient.java

C:\Users\Gagan\OneDrive\Documents\lekha_naik\RMI\CalcRMI>start rmiregistry

C:\Users\Gagan\OneDrive\Documents\lekha_naik\RMI\CalcRMI>java CalcRegister
```

```
C:\Windows\System32\cmd.exe - java CalcClient
Microsoft Windows [Version 10.0.22504.1010]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Gagan\OneDrive\Documents\lekha_naik\RMI\CalcRMI>java CalcClient
Result: 57.0
Result: -11.0
Result: 782.0
Result: 0.6764705882352942
```

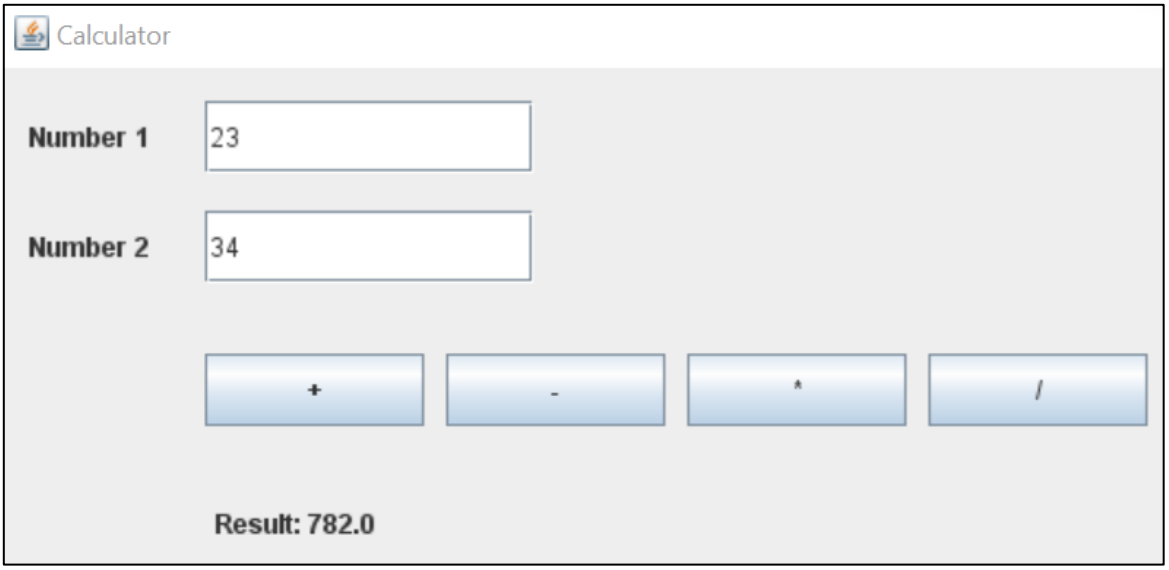
Addition:



Subtraction:



Multiplication:



Division: