Practical No.: 3A

Aim: Write a Program using RMI for Graphical User Interface(GUI) based Calculator

File → New Project → Java Application →Project Name: ***RMICalculator*** *→ Finish*

Right Click on Package Name (rmicalculator) → New → Java Interface → Class Name: ***CalculatorInterface***

CalculatorInterface.java

```
package rmicalculator;

import java.rmi.Remote;
import java.rmi.RemoteException;

/**
 *
 * @author TechnoBoy
 */
public interface CalculatorInterface extends Remote{
   public int add(int a,int b) throws RemoteException;
   public int sub(int a,int b) throws RemoteException;
   public int mul(int a,int b) throws RemoteException;
   public int div(int a,int b) throws RemoteException;
}
```

Right Click on Package Name (rmicalculator) → New → Java Class → Class Name: ***CalculatorInterfaceImpl***

CalculatorInterfaceImpl.java

```
package rmicalculator;

import java.rmi.RemoteException;
import java.rmi.server.UnicastRemoteObject;

/**
 *
 * @author TechnoBoy
 */
public class CalculatorInterfaceImpl extends UnicastRemoteObject implements CalculatorInterface{

   public CalculatorInterfaceImpl() throws RemoteException {
   }
   @Override
   public int add(int a, int b) throws RemoteException {
      return a+b;
   }
   @Override
   public int sub(int a, int b) throws RemoteException {
      return a-b;
   }
```

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

Right Click on Package Name (rmicalculator) → New → Java Class → Class Name: ***Server***

Server.java

```
package rmicalculator;

import java.rmi.RemoteException;
import java.rmi.registry.LocateRegistry;
import java.rmi.registry.Registry;
import java.util.logging.Level;
import java.util.logging.Logger;

/**
 *
 * @author TechnoBoy
 */
public class Server {
    public static void main(String[] args) {
        try {
            Registry cal = LocateRegistry.createRegistry(5000);
            cal.rebind("calculate", new CalculatorInterfaceImpl());
             System.out.println("Server Ready");
        } catch (RemoteException ex) {
            Logger.getLogger(Server.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
}
```

Right Click on Package Name (rmicalculator) → New → Java Class → Class Name: ***Client***

Client.java

```
package rmicalculator;

import java.net.MalformedURLException;
import java.rmi.Naming;
import java.rmi.NotBoundException;
import java.rmi.RemoteException;
import java.util.logging.Level;
import java.util.logging.Logger;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

/**
 *
 * @author TechnoBoy
```

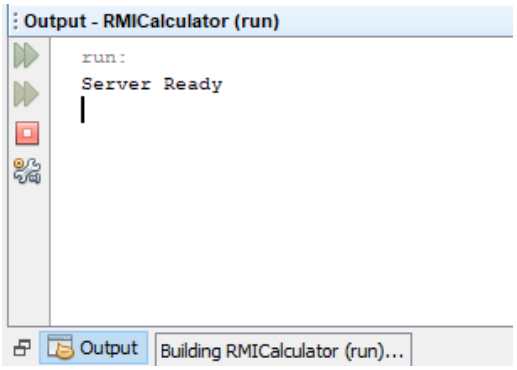
```
 */
public class Client {
    String strNum1 = "", strNum2 = "", strRes = "", op = "";
    double x, y, result;
    boolean flag, dotFlag, resFlag;
    CalculatorInterface ci;
    GridBagConstraints gbc = new GridBagConstraints();
    JTextField txt1 = new JTextField(20);
    JButton btn[] = new JButton[17];
    int i, j, k;
    Container con;

    public Client()
    {
        con = this.getContentPane();
        con.setLayout(new GridBagLayout());
        gbc.weightx = 1.0;
        gbc.weighty = 1.0;
        btn[0] = new JButton("C");
        btn[1] = new JButton("1");
        btn[2] = new JButton("2");
        btn[3] = new JButton("3");
        btn[4] = new JButton("+");
        btn[5] = new JButton("4");
        btn[6] = new JButton("5");
        btn[7] = new JButton("6");
        btn[8] = new JButton("-");
        btn[9] = new JButton("7");
        btn[10] = new JButton("8");
        btn[11] = new JButton("9");
        btn[12] = new JButton("*");
        btn[13] = new JButton("0");
        btn[14] = new JButton(".");
        btn[15] = new JButton("=");
        btn[16] = new JButton("/");
        gbc.gridx = 0;
        gbc.gridy = 0;
        gbc.gridwidth = 4;
        con.add(txt1, gbc);
        gbc.gridwidth = 1;
        gbc.gridx = 0;
        gbc.gridy = 1;
        con.add(btn[0], gbc);
        btn[0].addActionListener((ActionListener) this);
        i = 1;
        for (k = 2; k <= 5; k++)
        {
            for (j = 0; j <= 3; j++)
            {
                gbc.gridx = j;
                gbc.gridy = k;
                con.add(btn[i], gbc);
                btn[i].addActionListener((ActionListener) this);
                i++;
            }
```

```
    }
    setSize(300, 300);
    setVisible(true);
}
public void actionPerformed(ActionEvent ae)
{

    try {
        ci=(CalculatorInterface)Naming.lookup("rmi://localhost:5000/calculate");
    } catch (NotBoundException | MalformedURLException | RemoteException ex) {
        Logger.getLogger(Client.class.getName()).log(Level.SEVERE, null, ex);
    }
    String cmd = ae.getActionCommand();
    if (cmd.equals("C"))
    {
        txt1.setText("");
        strNum1 = strNum2 = strRes = "";
        x = y = result = 0;
        flag = true;
        dotFlag = false;
        resFlag = false;
    }
    else if (cmd.equals("+") || cmd.equals("-") || cmd.equals("*") || cmd.equals("/"))
    {
        flag = true;
        if (flag)
        {
            strNum1 = txt1.getText();
            strNum2 = "";
            x = Double.parseDouble(strNum1);
            flag = false;
            dotFlag = false;
            resFlag = false;
        }
        txt1.setText("");
        op = cmd;
    }
    else if (cmd.equals("="))
    {

        strNum2 = txt1.getText();
        y = Double.parseDouble(strNum2);
        try
        {
            if (op.equals("+")) result = ci.add(x, y);
            if (op.equals("-")) result = ci.sub(x, y);
            if (op.equals("*")) result = ci.mul(x, y);
            if (op.equals("/")) result = ci.div(x, y);
        }
        catch (Exception e)
        {
            e.printStackTrace();
        }
        txt1.setText(Double.toString(result));
        dotFlag = true;
```

```
            resFlag = true;
            flag = true;
        }
        else if (cmd.equals("."))
        {
            if (!dotFlag)
            {
                txt1.setText(txt1.getText() + cmd);
                dotFlag = true;
            }
        }
        else
        {
            if (!resFlag)
            {
                txt1.setText(txt1.getText() + cmd);
            }
        }
    }
    public static void main(String[] args) {
        Client cc=new Client();
    }
}
```

Output:

Run Server



Run client