

Unit 3

Q1) The client should provide an equation to the server through an interface. The server will solve the expression given by the client.

ICalculator.java:

```
import java.rmi.*;
public interface ICalculator extends Remote
{
    double add(double x,double y) throws RemoteException;
    double sub(double x,double y) throws RemoteException;
    double mul(double x,double y) throws RemoteException;
    double div(double x,double y) throws RemoteException;
}
```

CalculatorImpl.java:

```
import java.rmi.*;
import java.rmi.server.*;
public class CalculatorImpl extends UnicastRemoteObject implements
ICalculator
{
    public CalculatorImpl() throws RemoteException
    {
    }
    public double add(double x,double y) throws RemoteException
    {
        return(x+y);
    }
    public double sub(double x,double y) throws RemoteException
    {
        return(x-y);
    }
    public double mul(double x,double y) throws RemoteException
    {return(x*y);
    }
    public double div(double x,double y) throws RemoteException
    {
        return(x/y);
    }
}
```

CalculatorServer.java:

```
import java.net.*;
import java.rmi.*;
public class CalculatorServer
{
    public static void main(String[] args)
    {
        try
        {
            CalculatorImpl ci=new CalculatorImpl();
            Naming.rebind("CalculatorServer",ci);
            System.out.println("Calculator Server is Ready");
        }
        catch(Exception e)
        {
            e.printStackTrace();
        }
    }
}
```

CalculatorClient.java:

```
import java.rmi.*;
import java.net.*;
import java.io.*;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
public class CalculatorClient extends JFrame implements ActionListener
{
    String strNum1="",strNum2="",strRes="",op="";
    double x,y,result;
    boolean flag,dotFlag,resFlag;
    ICalculator intf;
    GridBagConstraints gbc=new GridBagConstraints();
    JTextField txt1=new JTextField(20);
    JButton btn[]=new JButton[17];
    int i,j,k;
```

```

Container con;
public CalculatorClient()
{
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(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(this);
i++;
}
}
setSize(300,300);
setVisible(true);
}
public void actionPerformed(ActionEvent ae)
{
try
{
String url="rmi://127.0.0.1/CalculatorServer";
intf=(ICalculator)Naming.lookup(url);
}
catch(Exception e)
{
e.printStackTrace();
}
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("/"))
{
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=intf.add(x,y);
}
if(op.equals("-"))
{
result=intf.sub(x,y);
}
if(op.equals("*"))
{
result=intf.mul(x,y);
}
if(op.equals("/"))
{
result=intf.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)
{
CalculatorClient cc=new CalculatorClient();
}
}
```

Output:

```
C:\Windows\System32\cmd.e  X  +  v  -  □  X

Microsoft Windows [Version 10.0.22631.4169]
(c) Microsoft Corporation. All rights reserved.

C:\DSCC>set path=C:\Program Files\Java\jdk1.8.0_241\bin

C:\DSCC>javac ICalculator.java

C:\DSCC>javac CalculatorImpl.java

C:\DSCC>start rmiregistry

C:\DSCC>|
```

```
C:\Windows\System32\cmd.e  X  +  v  -  □  X

Microsoft Windows [Version 10.0.22631.4169]
(c) Microsoft Corporation. All rights reserved.

C:\DSCC>set path=C:\Program Files\Java\jdk1.8.0_241\bin

C:\DSCC>javac CalculatorServer.java

C:\DSCC>java CalculatorServer
Calculator Server is Ready
|
```

```
C:\Windows\System32\cmd.e  X  +  v  -  □  X

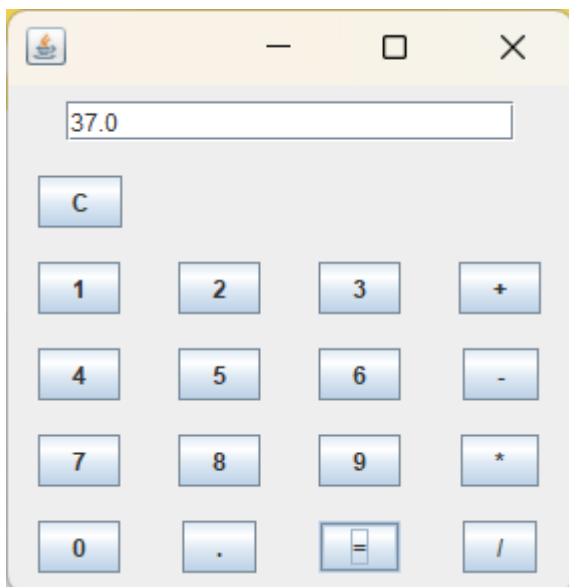
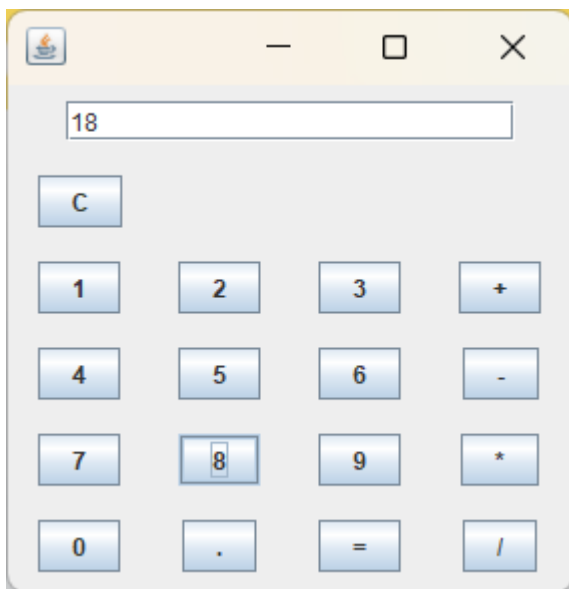
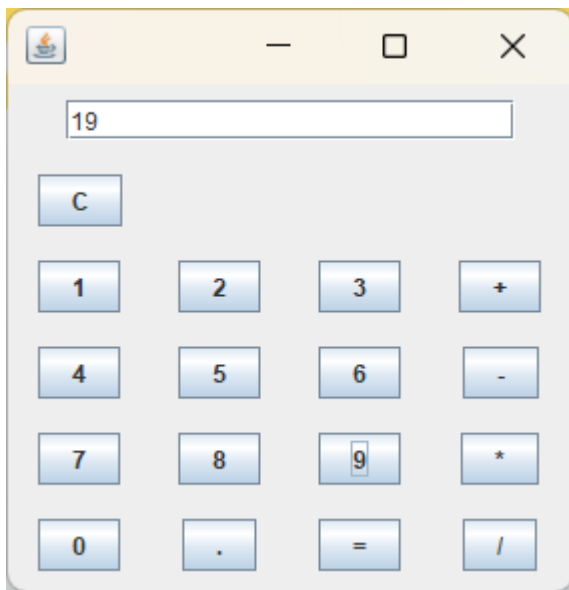
Microsoft Windows [Version 10.0.22631.4169]
(c) Microsoft Corporation. All rights reserved.

C:\DSCC>set path=C:\Program Files\Java\jdk1.8.0_241\bin

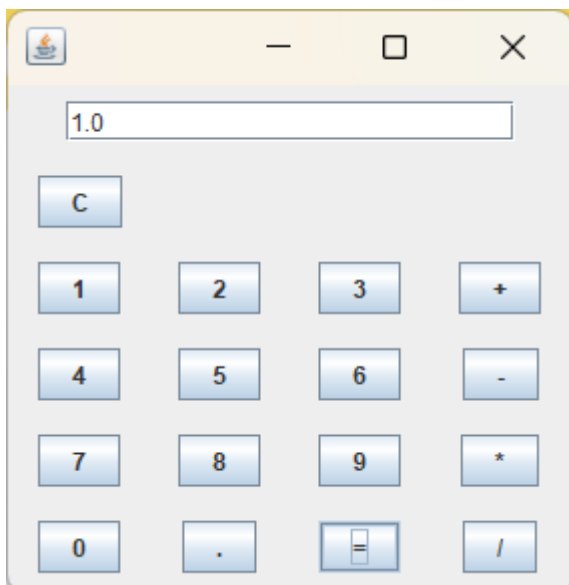
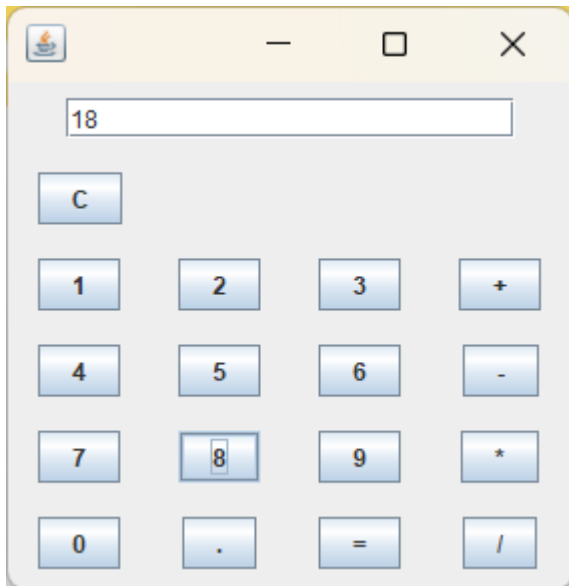
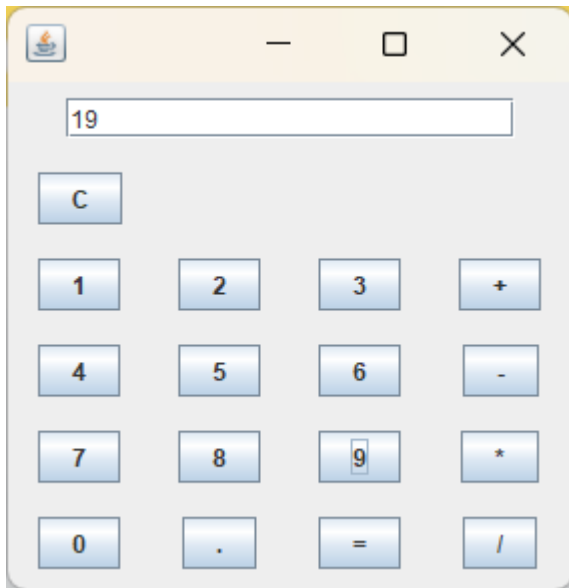
C:\DSCC>javac CalculatorClient.java

C:\DSCC>java CalculatorClient
```

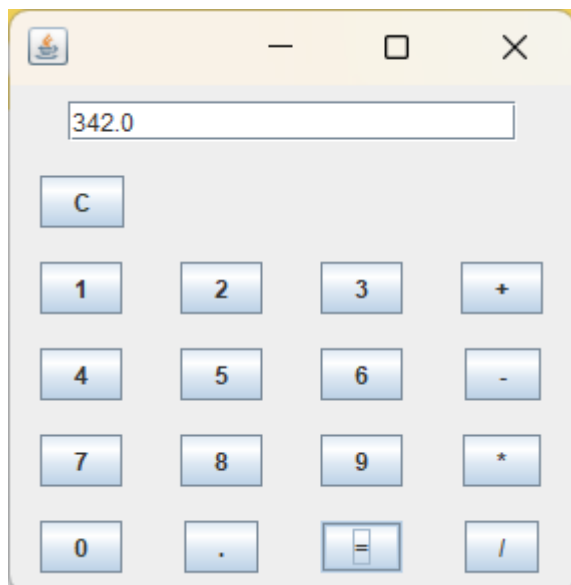
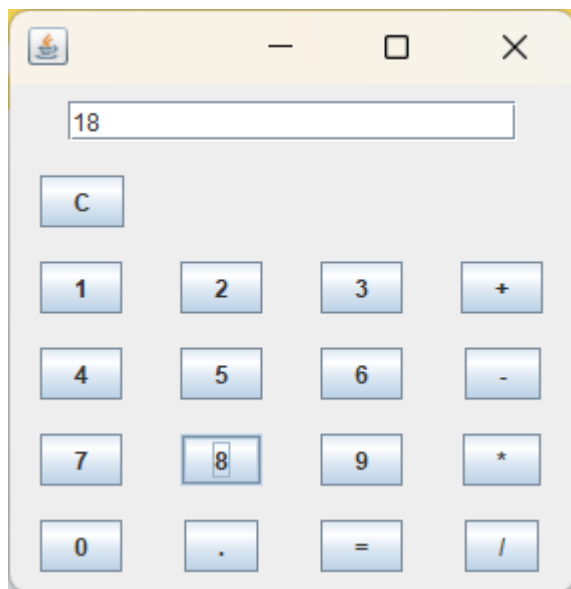
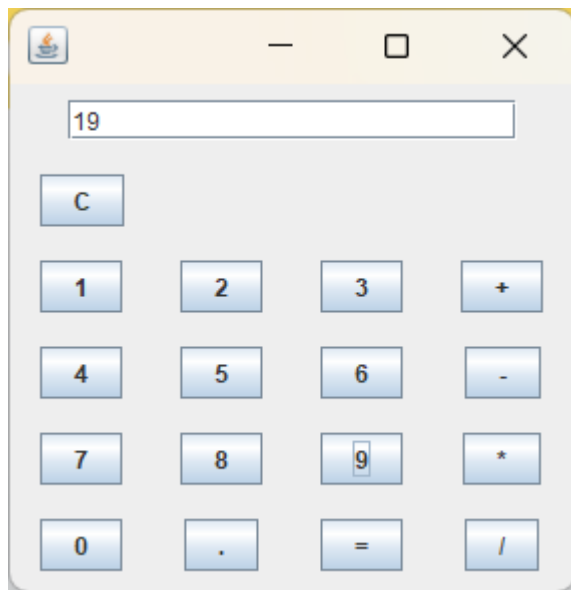

Add:



Subtract:



Multiply:



Divide:

