1. **Design an application to implement RMI in Java. Create a client program to invoke various Simple Metrics Calculator based methods on Server. Before passing the arguments to the server, the client application should accept inputs for the from and to unit names as well as the values for the same.**

**Code:**

**Client.java**

import java.rmi.\*;

public class Client {

public static void main(String args[]) {

try{

AddServerInterface st = (AddServerInterface)Naming.*lookup*("rmi://localhost/AddService");

System.*out*.println(st.sum(25,8));

}

catch(Exception e) {

System.*out*.println(e);

}

}

}

AddServerInterface.java

import java.rmi.\*;

public interface AddServerInterface extends Remote

{

public int sum(int a,int b) throws RemoteException;

}

**Adder.java**

import java.rmi.\*;

import java.rmi.server.\*;

public class Adder extends UnicastRemoteObject implements AddServerInterface

{

Adder() throws RemoteException

{

super();

}

public int sum(int a,int b)

{

return a+b;

}

}

**AddServer.java**

import java.rmi.\*;

import java.rmi.registry.\*;

public class AddServer {

public static void main(String args[]) {

try {

AddServerInterface addService=new Adder();

Naming.rebind("AddService",addService); //addService object is hosted with name AddService

}

catch(Exception e) {

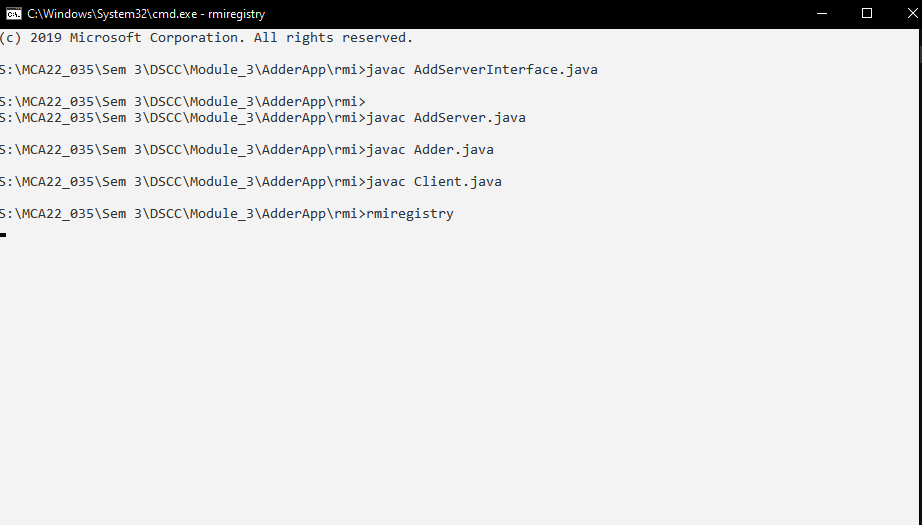
System.out.println(e);

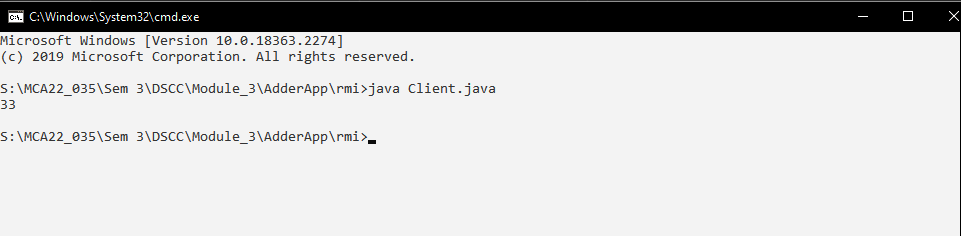
}

}

}

**Output:**





1. **Design a GUI application in java to implement RMI based simple calculator.**

**Code:**

**ICalculator.java**

import java.rmi.\*;

public interface ICalculator extends Remote {

int addOpr(int x,int y) throws RemoteException;

int subOpr(int x,int y) throws RemoteException;

int multOpr(int x,int y) throws RemoteException;

int divOpr(int x,int y) throws RemoteException;

}

**CalculatorImplementor.java**

import java.rmi.\*;

import java.rmi.server.\*;

public class CalculatorImplementor extends UnicastRemoteObject implements ICalculator

{

CalculatorImplementor() throws RemoteException {

super();

}

public int addOpr(int x,int y) throws RemoteException

{

return(x+y);

}

public int subOpr(int x,int y) throws RemoteException

{

return(x-y);

}

public int multOpr(int x,int y) throws RemoteException

{

return(x\*y);

}

public int divOpr(int x,int y) throws RemoteException

{

return(x/y);

}

}

**CalculatorServer.java**

import java.net.\*;

import java.rmi.\*;

public class CalculatorServer {

public static void main(String[] args)

{

try

{

ICalculator ical=new CalculatorImplementor();

Naming.rebind("calcservice",ical);

System.out.println("Calculator Server is Ready");

}

catch(Exception e)

{

e.printStackTrace();

}

}

}

**CalculatorClient.java**

import java.awt.\*;

import javax.swing.\*;

import java.rmi.\*;

import java.awt.event.\*;

public class CalculatorClient extends JFrame implements ActionListener

{

JButton b10,b11,b12,b13,b14,b15;

JButton b[]=new JButton[10];

int i,r,n1,n2;

JTextField res;

char op;

ICalculator ic;

public CalculatorClient()

{

super("CalculatorClient");

setLayout(new BorderLayout());

JPanel p=new JPanel();

p.setLayout(new GridLayout(4,4));

this.setDefaultCloseOperation(CalculatorClient.EXIT\_ON\_CLOSE);

for(int i=0;i<=9;i++)

{

b[i]=new JButton(i+"");

p.add(b[i]);

b[i].addActionListener(this);

}

b10=new JButton("+");

p.add(b10);

b10.addActionListener(this);

b11=new JButton("-");

p.add(b11);

b11.addActionListener(this);

b12=new JButton("\*");

p.add(b12);

b12.addActionListener(this);

b13=new JButton("/");

p.add(b13);

b13.addActionListener(this);

b14=new JButton("=");

p.add(b14);

b14.addActionListener(this);

b15=new JButton("C");

p.add(b15);

b15.addActionListener(this);

res=new JTextField(10);

add(p,BorderLayout.CENTER);

add(res,BorderLayout.NORTH);

setVisible(true);

setSize(200,200);

}

public void actionPerformed(ActionEvent ae)

{

try

{

String url="rmi://127.0.0.1/calcservice";

ic=(ICalculator)Naming.lookup(url);

}

catch(Exception e)

{

e.printStackTrace();

}

JButton pb=(JButton)ae.getSource();

if(pb==b15)

{

r=n1=n2=0;

res.setText("");

}

else if(pb==b14)

{

n2=Integer.parseInt(res.getText());

try {

eval();

} catch (RemoteException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

res.setText(""+r);

}

else

{

boolean opf=false;

if(pb==b10)

{

op='+';

opf=true;

}

if(pb==b11)

{

op='-';

opf=true;

}

if(pb==b12)

{

op='\*';

opf=true;

}

if(pb==b13)

{

op='/';

opf=true;

}

if(opf==false)

{

for(i=0;i<10;i++)

{

if(pb==b[i])

{

String t=res.getText();

t+=i;

res.setText(t);

}

}

}

else

{

n1=Integer.parseInt(res.getText());

res.setText("");

}

}

}//action performed

int eval() throws RemoteException

{

switch(op)

{

case '+':r=ic.addOpr(n1,n2);

break;

case '-':r=ic.subOpr(n1,n2);

break;

case '\*':r=ic.multOpr(n1,n2);

break;

case '/':r=ic.divOpr(n1,n2);

break;

}

return 0;

}

public static void main(String arg[])

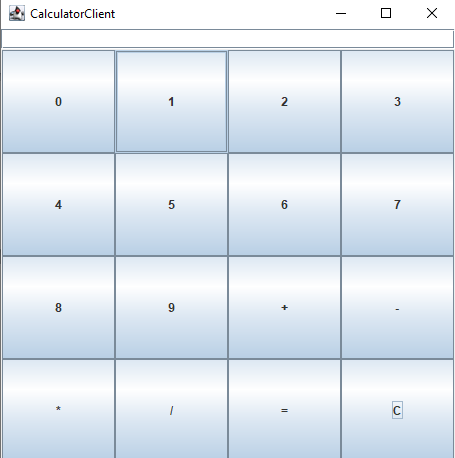
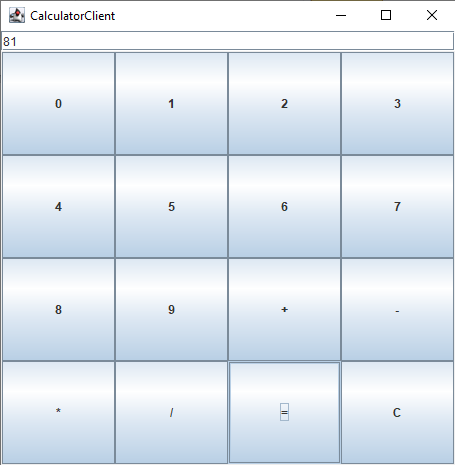
{

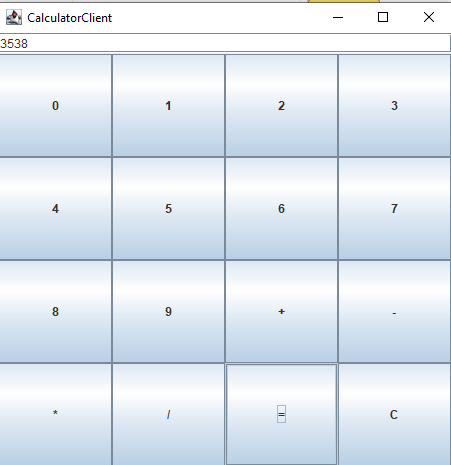
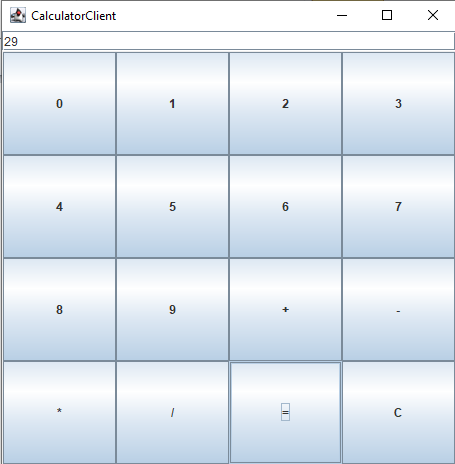
new CalculatorClient();

}

}

**Output:**

1. **Design an application using RMI in java to retrieve the Date and Time values from the server to the client program and display it**

**Code:**

**IDateTime.java**

import java.rmi.Remote;

import java.rmi.RemoteException;

public interface IDateTime extends Remote {

String getCurrentDateTime() throws RemoteException;

}

**DateTimeImplementer.java**

import java.rmi.RemoteException;

import java.rmi.server.UnicastRemoteObject;

import java.time.LocalDateTime;

import java.time.format.DateTimeFormatter;

public class DateTimeImplementer extends UnicastRemoteObject implements IDateTime {

public DateTimeImplementer() throws RemoteException{

super();

}

public String getCurrentDateTime() throws RemoteException {

DateTimeFormatter dtf = DateTimeFormatter.*ofPattern*("yyyy/MM/dd HH:mm:ss");

LocalDateTime now = LocalDateTime.*now*();

return String.*valueOf*(now);

}

}

**DateTimeServer.java**

import java.rmi.Naming;

public class DateTimeServer {

public static void main(String[] args) {

try {

IDateTime idate = new DateTimeImplementer();

Naming.*rebind*("DateTime", idate);

System.*out*.println("DateTime Server is Ready");

}catch (Exception e) {

System.*out*.println(e);

}

}

}

**DateTimeClient.java**

import java.rmi.Naming;

public class DateTimeClient {

public static void main(String[] args) {

try {

IDateTime service = (IDateTime)Naming.*lookup*("rmi://localhost/DateTime");

System.*out*.println(service.getCurrentDateTime());

}

catch (Exception e) {

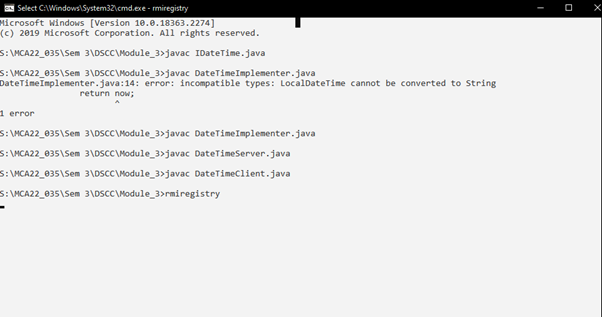
System.*out*.println(e);

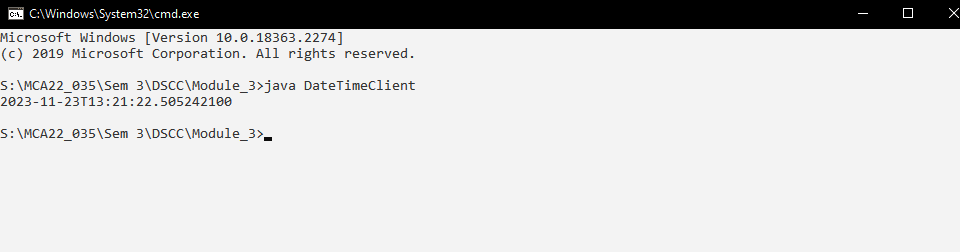
}

}

}

**Output:**





1. **Design a Equation solver application in java rmi to demonstrate the working of Pythagoras Theorem.**

**Code:**

**IPytho.java**

import java.rmi.Remote;

import java.rmi.RemoteException;

public interface IPytho extends Remote {

double getThirdSideLength(Double side1, Double side2) throws RemoteException;

}

**PythoImplementer.java**

import java.rmi.RemoteException;

import java.rmi.server.UnicastRemoteObject;

public class PythoImplementer extends UnicastRemoteObject implements IPytho {

public PythoImplementer() throws RemoteException {

super();

}

public double getThirdSideLength(Double side1, Double side2) {

double side3 = Math.sqrt(Math.pow(side1, 2) + Math.pow(side2, 2));

return side3;

}

}

**PythoServer.java**

import java.rmi.Naming;

public class PythoServer {

public static void main(String[] args) {

try {

IPytho ipy = new PythoImplementer();

Naming.*rebind*("Pytho", ipy);

System.*out*.println("Pytho service is ready");

}

catch (Exception e) {

System.*out*.println(e);

}

}

}

**PythoClient.java**

import java.rmi.Naming;

public class PythoClient {

public static void main(String[] args) {

try {

IPytho service = (IPytho)Naming.*lookup*("rmi://localhost/Pytho");

System.*out*.println(service.getThirdSideLength(4.5, 6.2));

}

catch (Exception e) {

System.*out*.println(e);

}

}

}

Output:

