

### Practical No.: 3C

**Aim: Write a Program using RMI to solve a Equation**

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

*Right Click on Package Name (rmiequation) → New → Java Interface → Class Name: Equation*

#### Equation.java

```
package rmiequation;

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

/**
 *
 * @author TechnoBoy
 */
public interface Equation extends Remote{
    public int Eq1(int a,int b) throws RemoteException;
    public int Eq2(int a,int b) throws RemoteException;
    public int Eq3(int a,int b) throws RemoteException;
    public int Eq4(int a,int b) throws RemoteException;
}
```

*Right Click on Package Name (rmiequation) → New → Java Class → Class Name: EquationImpl*

#### EquationImpl.java

```
package rmiequation;

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

/**
 *
 * @author TechnoBoy
 */
public class EquationImpl extends UnicastRemoteObject implements Equation {

    int ans=0;
    public EquationImpl() throws RemoteException {
    }
    @Override
    public int Eq1(int a, int b) throws RemoteException {
        ans=(a*a)+(2*a*b)+(b*b);
        return ans;
    }
    @Override
    public int Eq2(int a, int b) throws RemoteException {
        ans=(a*a)-(2*a*b)+(b*b);
        return ans;
    }
    @Override
```

```

public int Eq3(int a, int b) throws RemoteException {
    ans=(a*a*a)+(3*a*a*b)+(3*a*b*b)+(b*b*b);
    return ans;
}
@Override
public int Eq4(int a, int b) throws RemoteException {
    ans=(a*a*a)-(3*a*a*b)-(3*a*b*b)-(b*b*b);
    return ans;
}
}

```

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

### **Server.java**

```

package rmiequation;

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 equ = LocateRegistry.createRegistry(5000);
            equ.rebind("solver", new EquationImpl());
            System.out.println("Server Ready!!!!");
        } catch (RemoteException ex) {
            Logger.getLogger(Server.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
}

```

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

### **Client.java**

```

package rmiequation;

import java.net.MalformedURLException;
import java.rmi.Naming;
import java.rmi.NotBoundException;
import java.rmi.RemoteException;
import java.util.Scanner;
import java.util.logging.Level;
import java.util.logging.Logger;

```

```

/**
 *
 * @author TechnoBoy
 */
public class Client {
    public static void main(String[] args) {
        int choice, answer = 0, a, b;
        Scanner sc = new Scanner(System.in);
        try {
            while(true){
                Equation equ = (Equation) Naming.lookup("rmi://localhost:5000/solver");
                System.out.println("Equations :");
                System.out.println("1. (a+b)^2 \n2. (a-b)^2 \n3. (a+b)^3 \n4. (a-b)^3");
                System.out.println("Select a Equation :");
                choice = sc.nextInt();
                System.out.print("Enter value of a :");
                a = sc.nextInt();
                System.out.print("Enter value of b :");
                b = sc.nextInt();
                switch(choice)
                {
                    case 1:
                        answer = equ.Eq1(a, b);
                        break;

                    case 2:
                        answer = equ.Eq2(a, b);
                        break;

                    case 3:
                        answer = equ.Eq3(a, b);
                        break;

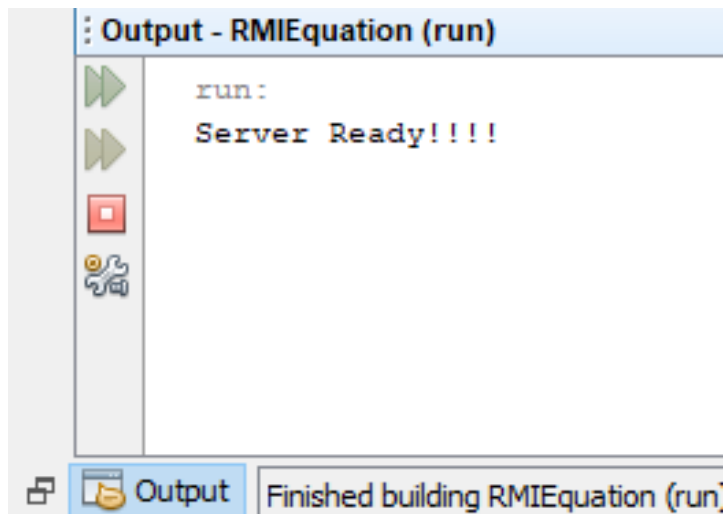
                    case 4:
                        answer = equ.Eq4(a, b);
                        break;

                    default:
                        System.out.println("Invalid Selection");
                }
                System.out.println("Answer : " + answer);
            }
        } catch (NotBoundException | MalformedURLException | RemoteException ex) {
            Logger.getLogger(Client.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
}

```

### Output:

#### *Run Server*



#### *Run Client*

