

NAME: GVS SAI MADHAV  
REG.NO: 19BCN7228  
SUBJECT: NETWORK PROGRAMMING  
LAB-1:

Serverinterface.java:

```
import java.rmi.*;
import java.rmi.server.*;
public interface servinterface extends Remote
{
    public double add(double a,double b) throws RemoteException;
    public double sub(double a,double b) throws RemoteException;
    public double mul(double a,double b) throws RemoteException;
    public double div(double a,double b) throws RemoteException;
    public double mod(double a,double b) throws RemoteException;
}
```

servimpl.java:

```
import java.rmi.*;
import java.rmi.server.*;
public class servimpl extends UnicastRemoteObject implements servinterface
{
    public servimpl() throws RemoteException
    {
        super();
    }
    public double add(double a,double b)
    {
        return(a+b);
    }
    public double sub(double a,double b)
    {
        return(a-b);
    }
    public double mul(double a,double b)
    {
        return(a*b);
    }
    public double div(double a,double b)
    {
        return(a/b);
    }

    public double mod(double a,double b)
    {
        return(a%b);
    }
}
```

## server.java

```
import java.rmi.*;
import java.rmi.server.*;
import java.rmi.registry.*;
public class server extends servimpl
{
    public server() throws RemoteException
    {
    }
    public static void main(String args[]) throws RemoteException
    {
        try
        {
            Registry reg = LocateRegistry.createRegistry(9999);
            servimpl s = new servimpl();
            reg.rebind("Server",s);
            System.out.println("Server is ready!");
        }
        catch(Exception e)
        {
            System.out.println(e);
        }
    }
}
```

## client.java

```
import java.rmi.*;
import java.rmi.server.*;
import java.rmi.registry.*;
public class client
{
    public static void main(String args[])throws RemoteException
    {
        try
        {
            int a = Integer.parseInt(args[0]);
            int b = Integer.parseInt(args[1]);
            Registry reg=LocateRegistry.getRegistry("127.0.0.1",9999);
            servinterface i = (servinterface)reg.lookup("Server");
            System.out.println("First number: "+a); System.out.println("Second
number: "+b);
            System.out.println("Add:\t"+i.add(a,b));
            System.out.println("Sub:\t"+i.sub(a,b));
            System.out.println("Mul:\t"+i.mul(a,b));
            System.out.println("Div:\t"+i.div(a,b));
            System.out.println("Mod:\t"+i.mod(a,b));
        }
        catch(Exception e)
        {
            System.out.println(e);
        }
    }
}
```

### Execution:

```
PS E:\Lab-Reports\NWP\Try> javac *.java
PS E:\Lab-Reports\NWP\Try> rmic servimpl
Warning: generation and use of skeletons and static stubs for JRMP
is deprecated. Skeletons are unnecessary, and static stubs have
been superseded by dynamically generated stubs. Users are
encouraged to migrate away from using rmic to generate skeletons and static
stubs. See the documentation for java.rmi.server.UnicastRemoteObject.
PS E:\Lab-Reports\NWP\Try> rmiregistry
```

### Starting the server:

```
PS E:\Lab-Reports\NWP\Try> java server
Server is ready!
```

### Executing the client:

```
PS E:\Lab-Reports\NWP\Try> java client 100 50
First number: 100
Second number: 50
Add: 150.0
Sub: 50.0
Mul: 5000.0
Div: 2.0
Mod: 0.0
PS E:\Lab-Reports\NWP\Try>
```