Assignment No.1

Implement multi-threaded client/server Process communication using RMI.

Problem Statement:

To develop any distributed application through implementing client-server communication

programs based on Java RMI .

Code:

1. **AddServerIntf.java**

```java
import java.rmi.*;
public interface AddServerIntf extends Remote {
double add(double d1, double d2) throws RemoteException;
}
```

2. **AddServerImpl.java**

```java
import java.rmi.*;
import java.rmi.server.*;
public class AddServerImpl extends UnicastRemoteObject
implements AddServerIntf {
public AddServerImpl() throws RemoteException {
}
public double add(double d1, double d2) throws RemoteException {
return d1 + d2;
}
}
```

3. **AddServer.java**

```java
import java.net.*;
import java.rmi.*;
public class AddServer {
public static void main(String args[]) {
try {
AddServerImpl addServerImpl = new AddServerImpl();
Naming.rebind("AddServer", addServerImpl);
}
catch(Exception e) {
System.out.println("Exception: " + e);
}
}
}
```

4. **AddClient.java**

```
import java.rmi.*;
public class AddClient {
public static void main(String args[]) {
try {
String addServerURL = "rmi://" + args[0] + "/AddServer";
AddServerIntf addServerIntf =
(AddServerIntf)Naming.lookup(addServerURL);
System.out.println("The first number is: " + args[1]);
double d1 = Double.valueOf(args[1]).doubleValue();
System.out.println("The second number is: " + args[2]);
double d2 = Double.valueOf(args[2]).doubleValue();
System.out.println("The sum is: " + addServerIntf.add(d1, d2));
}
catch(Exception e) {
System.out.println("Exception: " + e);
}
}
}
```

OUTPUT:


```
student@ioe-it-lab-2-129: ~/Desktop/BE_Hetavi(19)/RMI

student@ioe-it-lab-2-129:~/Desktop/BE_Hetavi(19)/RMI$ javac *.java
student@ioe-it-lab-2-129:~/Desktop/BE_Hetavi(19)/RMI$ rmic AddServerImpl
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been superseded by dynamically generated stubs.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.
student@ioe-it-lab-2-129:~/Desktop/BE_Hetavi(19)/RMI$ rmiregistry
```



```
student@ioe-it-lab-2-129: ~/Desktop/BE_Hetavi(19)/RMI/server

student@ioe-it-lab-2-129:~/Desktop/BE_Hetavi(19)/RMI/server$ java AddServer
```

```
student@ioe-it-lab-2-129:~/Desktop/BE_Hetavi(19)/RMI/client$ java AddClient 127.0.0.1 45 19
The first number is: 45
The second number is: 19
The sum is: 64.0
student@ioe-it-lab-2-129:~/Desktop/BE_Hetavi(19)/RMI/client$
```