**Drawback in Socket Programming:**To develop the Distributed application, if we use socket programming then programmer has to write statements for creating client side socket ,serverside socket, inputStream, and outputstream at local machine and remote machines. This is burden to developer.

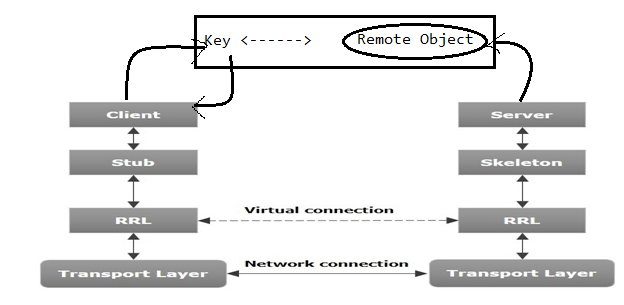
To over come the above problem, SUN Microsystem has provides another and new technology which is RMI.

In case of RMI , the complete distributed applications infrastructure like serverside socket, client side socket, input stream and Output Streams are provided by RMI Internally. Programmer does not need to write statements for creating them.

**1.RMI Application:-** server. A client is an application that requires services of object . A server creates an object and makes object available to clients. The object is called a remote object. This remote object contains remote method. A client contacts server to reference and invoke object by using RMI.

A RMI is used to build distributive applications.

Note:- The Client and Server side objects both are written in JAVA.



1.1)Working of RMI Application:-

* When the client makes a call to the remote object, it is received by the stub which eventually passes this request to the RRL. The stub is (Proxy)representation of object at client side.
* The RRL on the server side passes the request to the Skeleton (proxy on the server) which finally invokes the required remote method on the server. The skeleton get value which is returned by remote method.
* The result is passed all the way back to the client.

**2. RMI Components:-**  The RMI will use following components.

a.Stub.

b.Skelton.

c. Registry Software.

**2.1.Stub:-**

* It resides at client Machine.
* It provides client side socket, Output stream, input stream.
* It performs serialization,deserilization,Marshalling and UnMarshalling.
* It receives remote method call and it will send to RRL.
* It receives values which came from skelton and It send data to client application.

**2.2.Skelton:-**

* It resides at server Machine.
* It provides server side socket, Output stream, input stream.
* It performs serialization,deserilization,Marshalling and UnMarshalling.
* It receives remote method from RRL.
* It invokes the remote method and it receives the value which is returned by the remote method.
* Skelton send value to RRL.

**2.3.RMI Registry:-** It is a Registry software. This software manages private area at server machine. After creating Remote object, server application places it in private area. This area contains the remote objects in the form of key/value pair.

The key is logical name of Remote object and value is remote object. The area and object in this area are managed by Registry software. RMI Registry make available them to all clients.

**3.Steps to Prepare RMI Application:-**

* **Create Remote Interface.**
* **Create Remote Interface implementation class.**
* **Create Registry program**
* **Create Client Application.**
* **Run RMI Application.**

**3.1. Create Remote Interface:-** The main intention of Remote Interface is to declare all remote services as abstract methods.

a. Declare the User defined Interface.

b. Extend java.rmi.Remote interface to user defned interfaces.

c. Declare the remote methods in the form of abstract methods . The methods throws the java.rmi.RemoteException

**3.2. Create Remote Interface implementation class:-**

1. Declare the User Defined class that has to implements Remote Interface.
2. Declare public and 0-arg constructor in user-defined class with throws java.rmi.RemoteException.
3. Provide the body to abstract methods of remote interface.

**3.3. Create Registry Program:-** This program creates Remote object and keep the remote object in RMI registry with logical name.

This program invoke either bind or rebind method. These methods are in java.rmi.Naming class.

Syntax:

Public static void bind/rebind(String logical\_name,Remote object) throws RemoteException

This method bind remote object with logical name . This method keep the remote object with logical name in RMI Registry.

**3.4 Create Client Application:-** The client application get Remote object from RMI Registry and It uses remote object to access the remote method.

This application invokes lookup method. This method is in java.rmi.Naming class.

Syntax:

Public static void lookup(String logicalname);

This method returns the remote object of specified logical name.

**3.5. Execute RMI application:-**

Step1: Opent two commond prompts. One for client and another one for server. Compile all .java files.

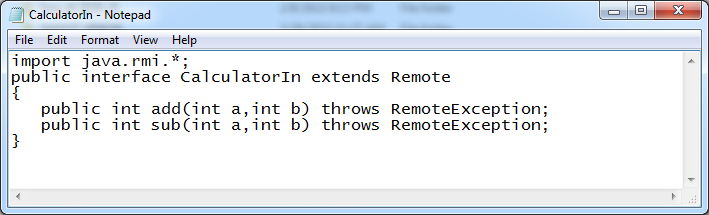
Step2: Start the RMI Registry.

Step3: Execute the RMI registry program.

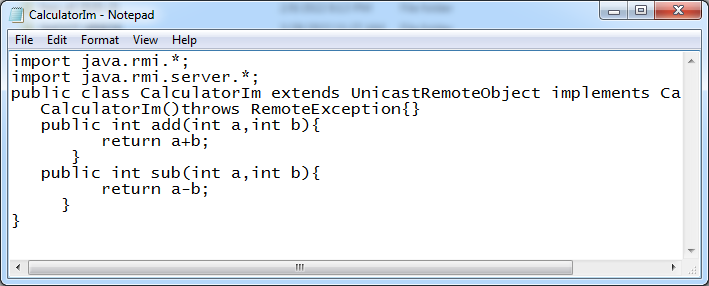
Step4: Execute the Client Application.

**CALCULATOR APPLICATION**

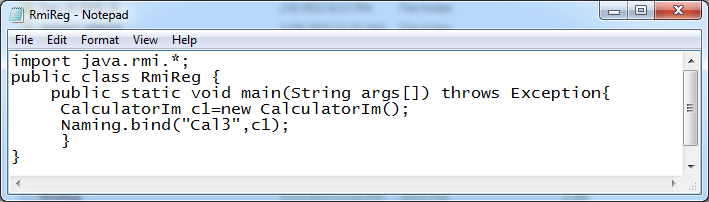
1. **CalculatorIn.java**

****

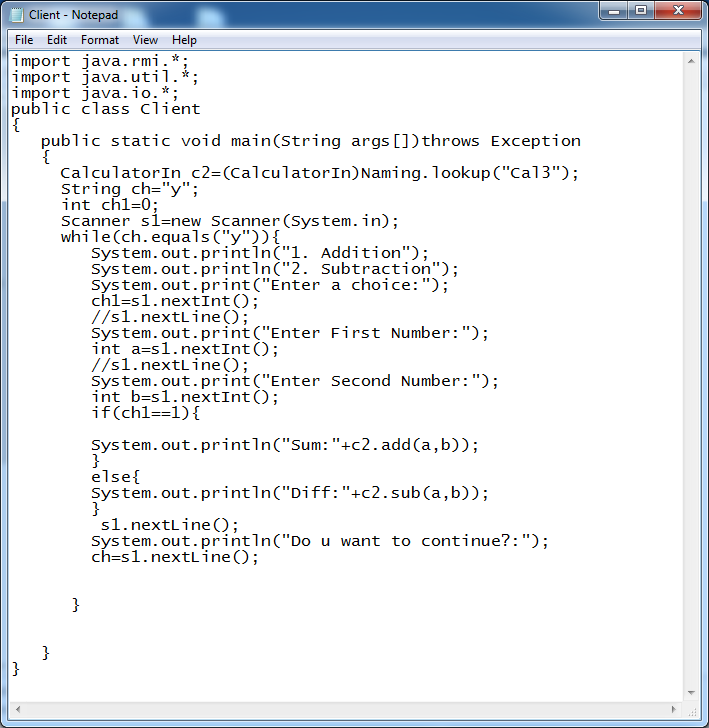
1. **CalculatorIm.java**

****

1. **RmiReg.java**

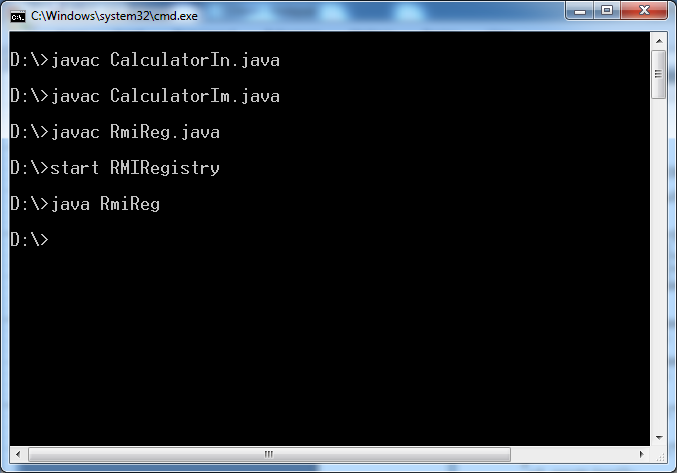
****

1. **Client.java**

****

**Output:**

**Server:**

****

Client:

