Lab#06

**Objective:** To learn RMI-IIOP

**Task:** Create RMI IIOP application in which take a sentence at runtime on the client application and let the server return its reverse.  
E.g. input could be: subject is distributed computing   
output should be: computing distributed is subject.

**Interface:**

import java.rmi.Remote;

public interface MessegeInterface extends Remote{

public String reverseMessege(String msg)throws java.rmi.RemoteException;

}

**RemoteObject Class:**

import javax.rmi.PortableRemoteObject;

import java.rmi.RemoteException;

public class MessegeImpl extends PortableRemoteObject implements MessegeInterface{

public MessegeImpl()throws RemoteException{

super();

}

public String reverseMessege(String msg)throws RemoteException{

String messege[]=msg.split(" ");

System.out.println("length"+messege.length);

String m=" ";

for(int i=messege.length-1; i>=0; i--){

m+=messege[i]+" ";

}

return m;

}

**}**

**ServerClass:**

import javax.naming.InitialContext;

import javax.naming.Context;

public class MessegeServer{

public static void main(String args[]){

try{

MessegeImpl impl=new MessegeImpl();

Context initailNamingContext=new InitialContext();

initailNamingContext.rebind("MessegeServer",impl);

System.out.println("Hello Server:Ready");

}catch(Exception e){

System.out.println(e);

}

}

}

**Client Class:**

import java.net.\*;

import java.rmi.\*;

import javax.rmi.PortableRemoteObject;

import java.util.\*;

import javax.naming.\*;

import java.io.\*;

public class ClientMessege{

public static void main(String args[]){

Context ic;

Object objref;

MessegeInterface mi;

try{

ic=new InitialContext();

objref=ic.lookup("MessegeServer");

System.out.println("Client:objtaind a ref.to MessegeServer");

mi=(MessegeInterface)PortableRemoteObject.narrow(objref,MessegeInterface.class);

System.out.println(mi.reverseMessege("Hi Suraj Kumar here"));

}catch(Exception e){

System.out.println(e);

}

}

}

**Output:**

