**LAB:NO:04**

**Object: To get familiar with Rmi call backs.**

**Task: Develop the RMI Callback application of Payroll and PayrollMessage services for calculating the earnings.**

**1)server Interface:**

import java.rmi.\*;

public interface payrollInterface extends Remote{

public double earnings(int id,double hours) throws RemoteException;

}

**2)Client Interface:**

import java.rmi.\*;

public interface payrollMessageInterface extends Remote{

public void message(String str) throws RemoteException;

}

**3)Server Remote object:**

import java.rmi.\*;

import java.rmi.server.\*;

public class payrollImpl extends UnicastRemoteObject implements payrollInterface{

private int id;

private double hours;

public payrollImpl() throws RemoteException{

}

public double earnings(int id,double hours)throws RemoteException{

try{

payrollMessageInterface pm = (payrollMessageInterface)Naming.lookup("rmi://localhost:1099/message");

pm.message("payroll client registry has been llokup by server");

}

catch(Exception e){

}

return id\*hours;

}

}

**4)Client Remote object:**

import java.rmi.\*;

import java.rmi.server.\*;

public class payrollMessageImpl extends UnicastRemoteObject implements payrollMessageInterface{

private String message;

public payrollMessageImpl() throws RemoteException{

}

public void message(String str)throws RemoteException{

System.out.println("hello");

}

}

**5)Server Main Class:**

import java.rmi.\*;

public class payrollServer{

public static void main(String args[]){

try{

String ro = "rmi://localhost:1099/money";

payrollImpl p = new payrollImpl();

Naming.rebind(ro,p);

System.out.println("Hello Server is ready");

System.out.println("Sending earning values to client");}

catch(Exception e){

}

}

}

**6)Client Main class:**

import java.rmi.\*;

class payrollClient{

public static void main(String args[]){

try{

System.out.println("Hello Client is ready");

Naming.rebind("rmi://localhost:1099/message",new payrollMessageImpl());

System.out.println("Receiving earnings values from server");

payrollInterface p = (payrollInterface)Naming.lookup("rmi://localhost:1099/money");

double m = p.earnings(7,5.5);

System.out.println(m);

}

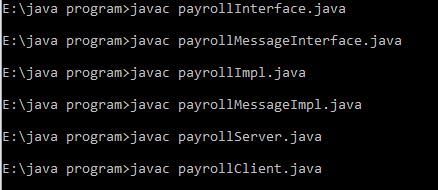
catch(Exception e){

}

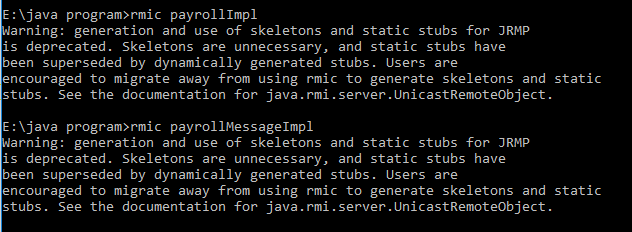
}

}

**Output: compile all classes**

****

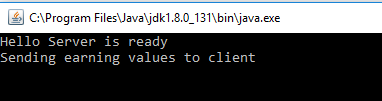
**Create skeleton and stub of server remote and client remote object.**

****

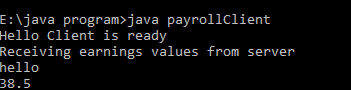
**Start registry to register client and server objects.**

****

**Run server main class:**

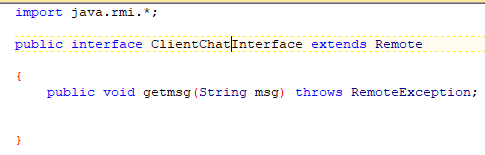
****

**Run client Main class:**

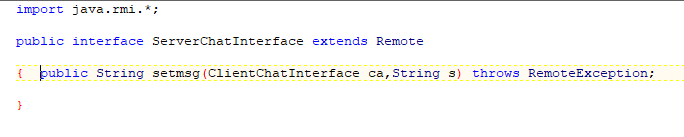
****

**Task 2:Optional Task: Develop an RMI application for chatting between client and server , the chatting should stop when both of them say “bye”.**

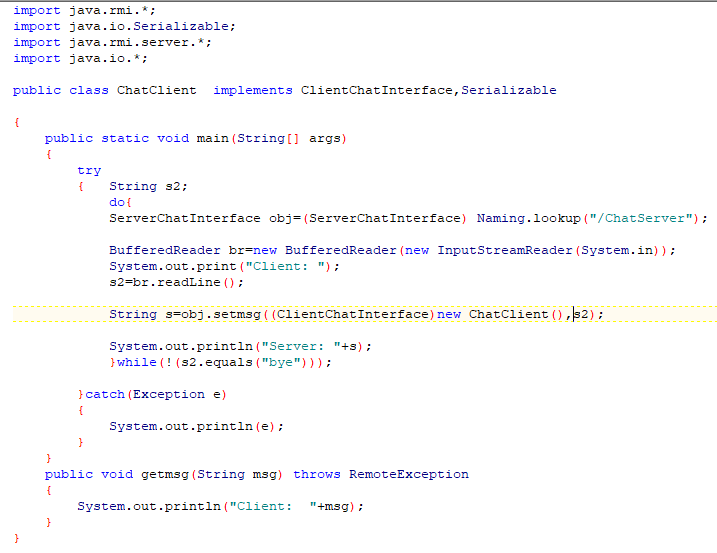
**Client interface:**

****

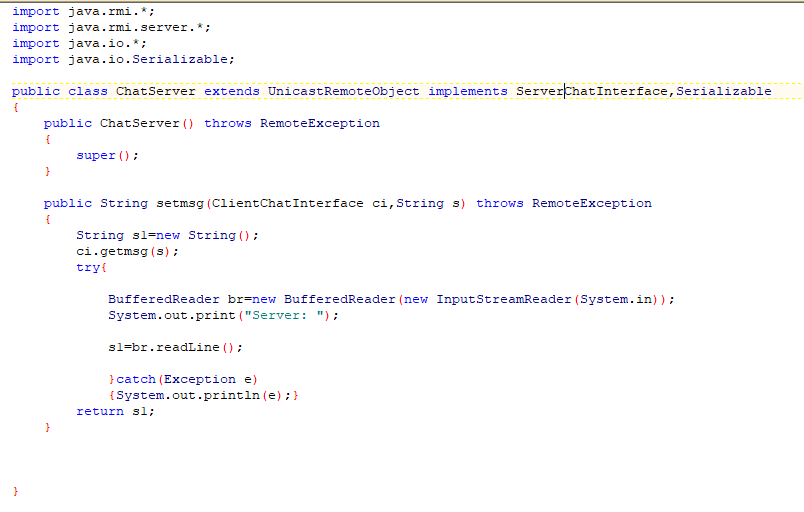
**Server interface:**

****

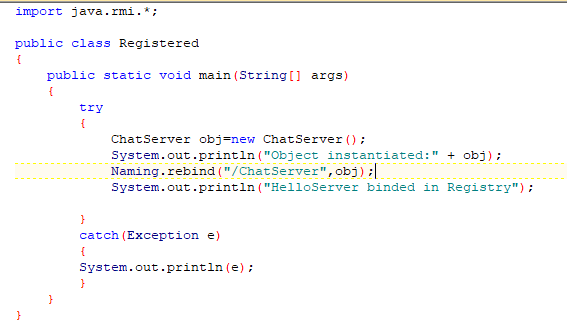
**Client Remote object class with main method:**

****

**Server Remote Object class:**

****

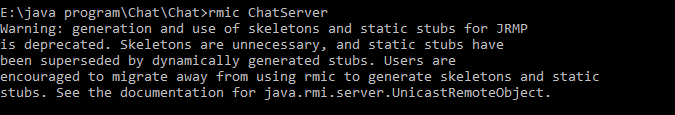
**Server main class:**

****

**Output:**

**First compile all the classes:**

**Then create skeleton of server remote object:**

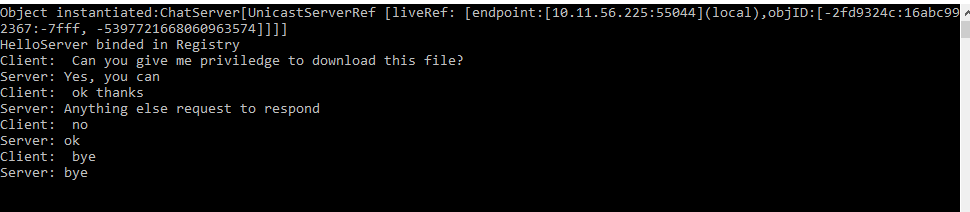
****

**Start registry to register the server remote object**

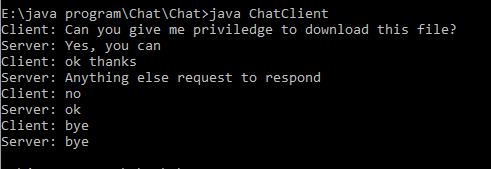
****

**Run server main class first**

****

****

**Then client main class**

****