Lab 4

* Develop the RMI Callback application of Payroll and PayrollMessage services for calculating the earnings.
* Optional Task: Develop an RMI application for chatting between client and server , the chatting should stop when both of them say “bye”.

Task 1.

**PayrollInterface.java**

import java.rmi.\*;

public interface payrollInterface extends Remote{

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

}

**payrollMessageInterface.java**

import java.rmi.\*;

public interface payrollMessageInterface extends Remote{

public void message(String str) throws RemoteException;

}

**payrollImpl.java**

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("client registry has been lookup by server");

}

catch(Exception e){

}

return id\*hours;

}

}

**payrollMessageImpl.java**

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("ahsan");

}}

**payrollServer.java**

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("Server is ready");

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

catch(Exception e){

}

}

}

**payrollClient.java**

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(6,9);

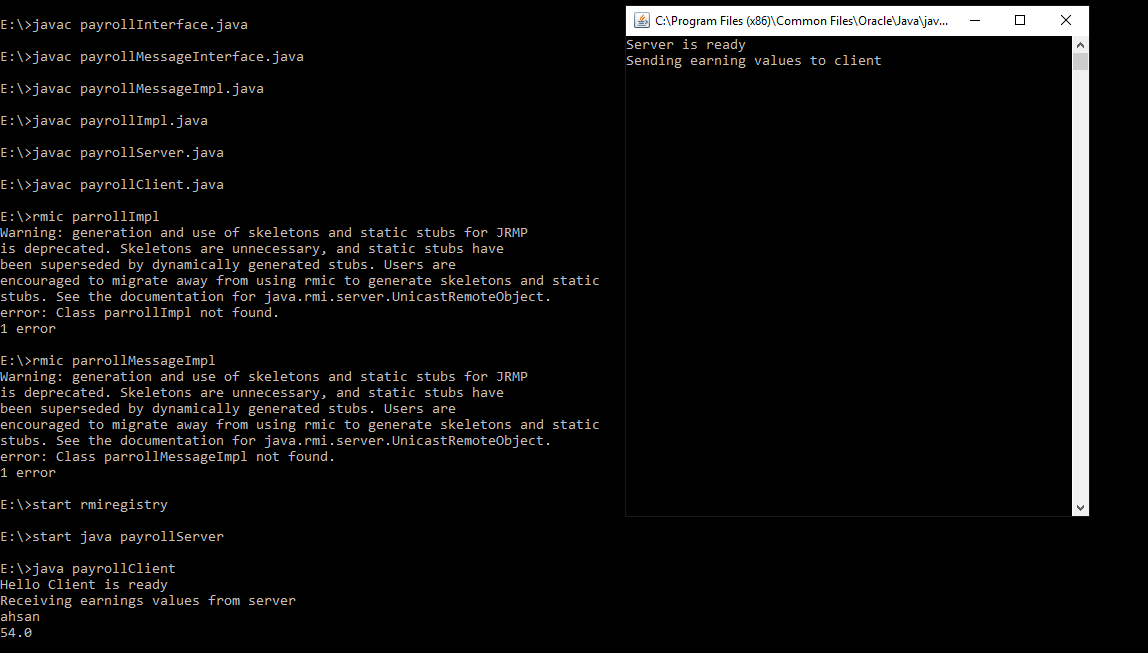
System.out.println(m);

}catch(Exception e){

}

}}

**Output:**



Task 2.

**ClientChatInterface.java**

import java.rmi.\*;

public interface ClientChatInterface extends Remote{

public void getmsg(String msg) throws RemoteException;

}

**ServerChatInterface.java**

import java.rmi.\*;

public interface ServerChatInterface extends Remote{

public String setmsg(ClientChatInterface ca,String s)throws RemoteException;

}

**ChatClient.java**

import java.rmi.\*;

import java.io.Serializable;

import java.rmi.server.\*;

import java.io.\*;

public class ChatClient implements ClientChatInterface,Serializable{

public static void main(String args[]){

try{

String s1;

do{

ServerChatInterface obj=(ServerChatInterface) Naming.lookup("/ChatServer");

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

System.out.println("Client:");

s1=br.readLine();

String s=obj.setmsg((ClientChatInterface)new ChatClient(),s1);

System.out.println("server:"+s);

}

while(!(s1.equals("bye")));

}catch(Exception e){

System.out.println(e);

}}

public void getmsg(String msg)throws RemoteException{

System.out.println("client:"+msg);

}}

**ChatServer.java**

import java.rmi.\*;

import java.rmi.server.\*;

import java.io.\*;

import java.io.Serializable;

public class ChatServer extends UnicastRemoteObject implements ServerChatInterface,Serializable{

public ChatServer()throws RemoteException{

super();

}

public String setmsg(ClientChatInterface ci,String s)throws RemoteException{

String s2=new String();

ci.getmsg(s);

try{

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

System.out.println("server");

s2=br.readLine();

}

catch(Exception e){

System.out.println(e);

}

return s2;

}}

**Reg.java**

import java.rmi.\*;

public class Reg{

public static void main(String args[]){

try{

ChatServer obj=new ChatServer();

System.out.println("obj instantiated:"+obj);

Naming.rebind("/ChatServer:",obj);

System.out.println("HelloServer binded in reg");

}catch(Exception e){

System.out.println(e);

}}

}

**OUTPUT :**

