Department of Software Engineering
Mehran University of Engineering and Technology, Jamshoro

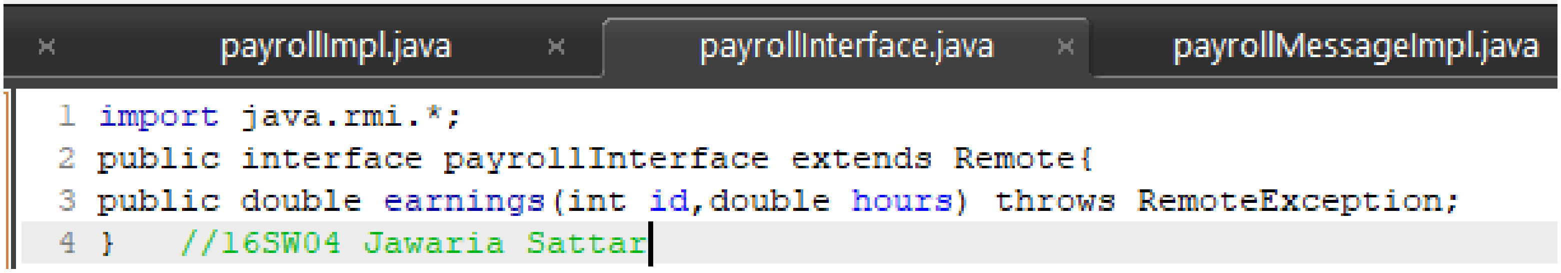
Course: SW422–Distributed Computing			
Instructor	RabeeaJaffari	**Practical/Lab No.**	04
Date		**CLOs**	CLO-3: P5 & CLO-3: P3
Signature		**Assessment Score**	

Topic	**To work with Remote Method Invocation (RMI) Callbacks**
Objectives	- Learn remote callbacks in distributed applications

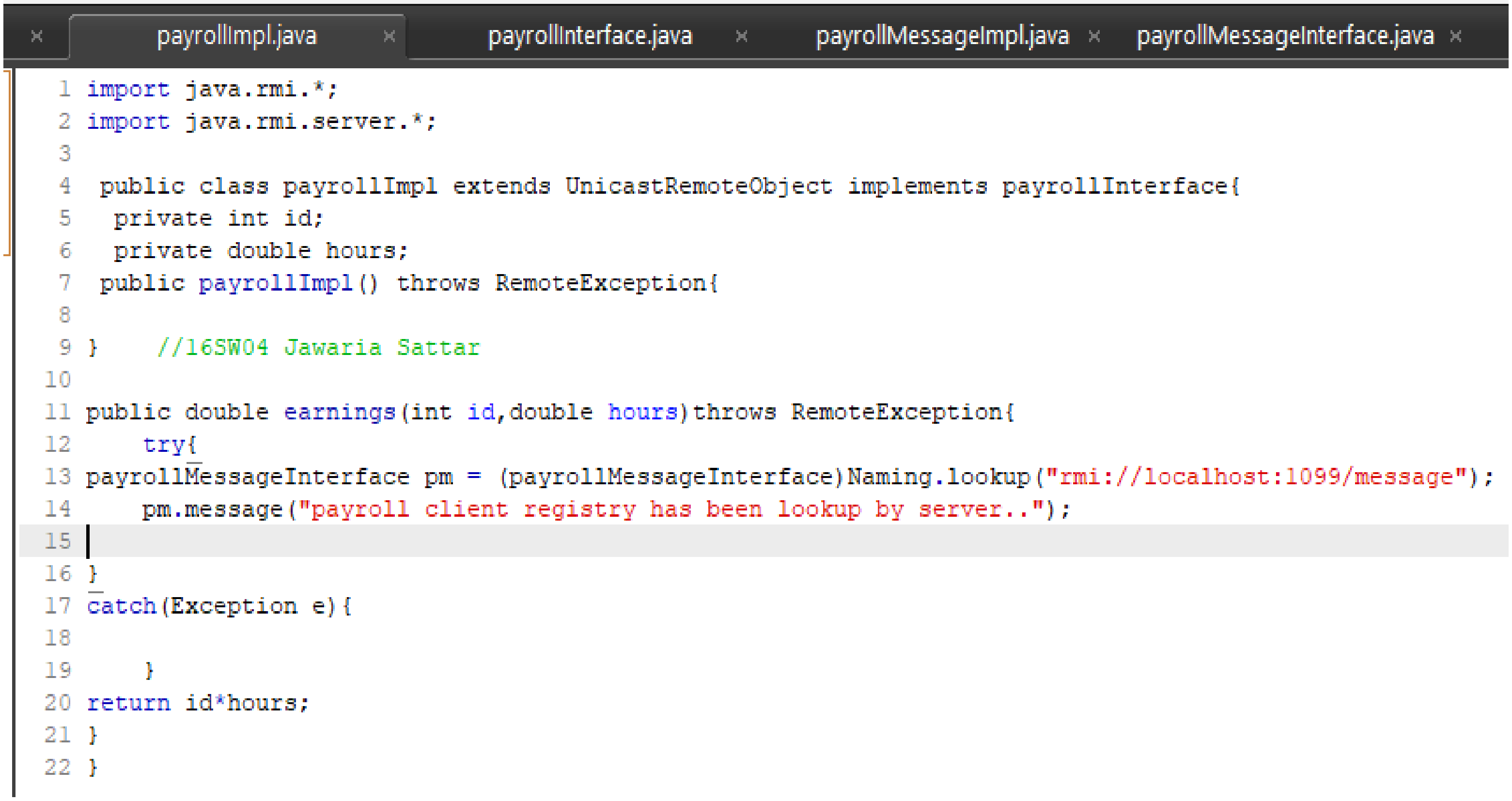
Lab Discussion: Theoretical concepts and Procedural steps

Lab Tasks
Submission Date:

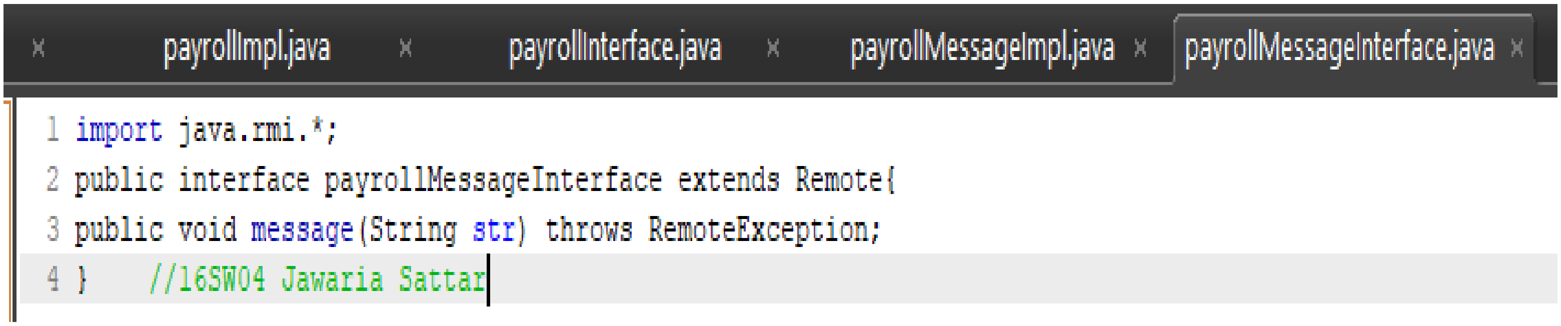
1. Develop the RMI Callback application of Payroll and PayrollMessage services for calculating the earnings.



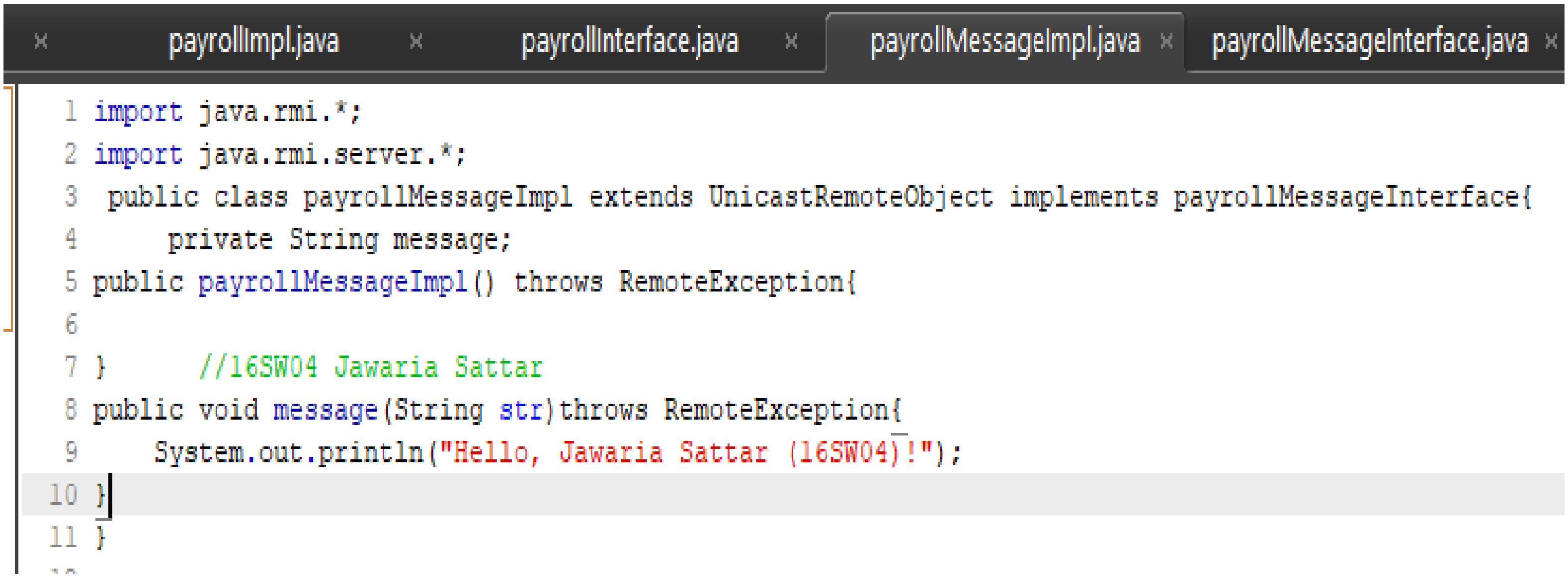
```
import java.rmi.*;
public interface payrollInterface extends Remote{
public double earnings(int id,double hours) throws RemoteException;
}    //16SW04 Jawaria Sattar
```



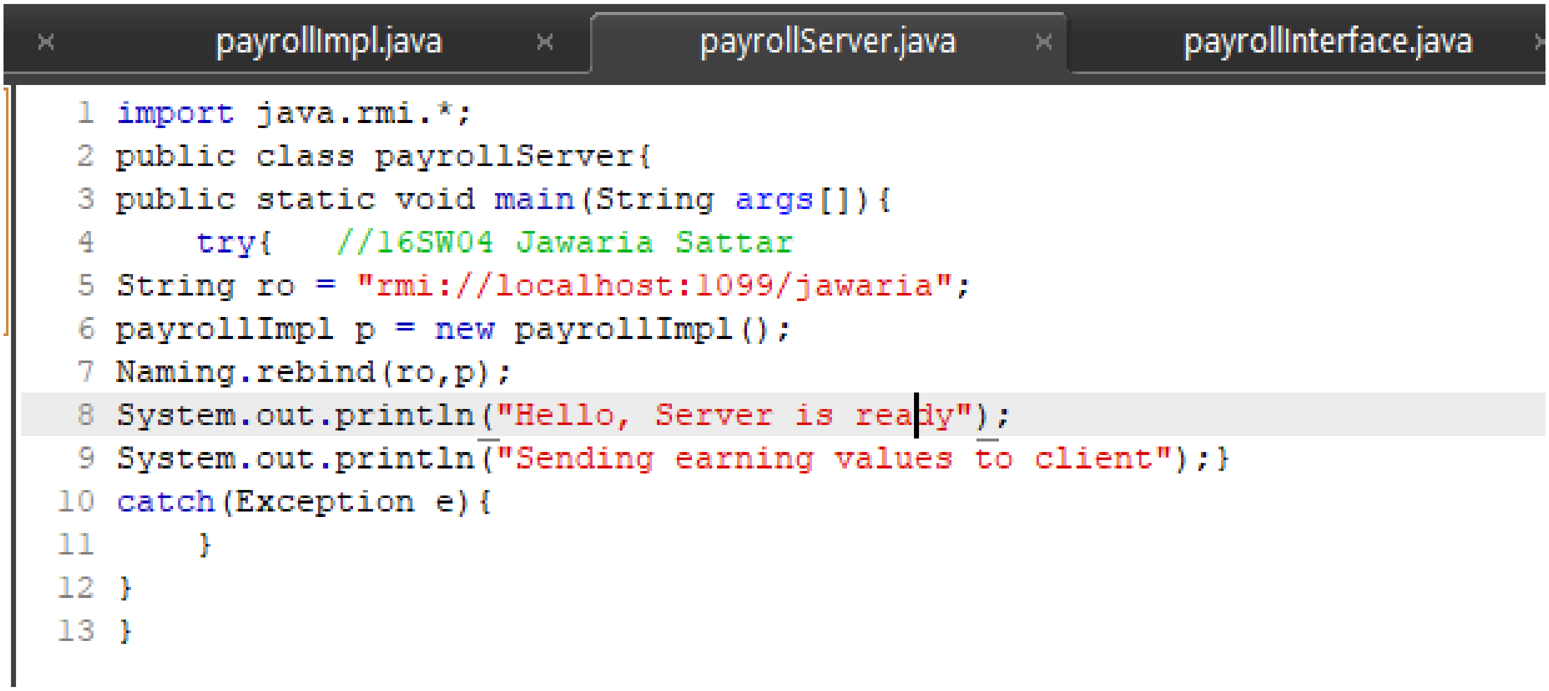
```
import java.rmi.*;
import java.rmi.server.*;

 public class payrollImpl extends UnicastRemoteObject implements payrollInterface{
  private int id;
  private double hours;
 public payrollImpl() throws RemoteException{

}    //16SW04 Jawaria Sattar

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 lookup by server..");

}
catch(Exception e){

    }
return id*hours;
}
}
```



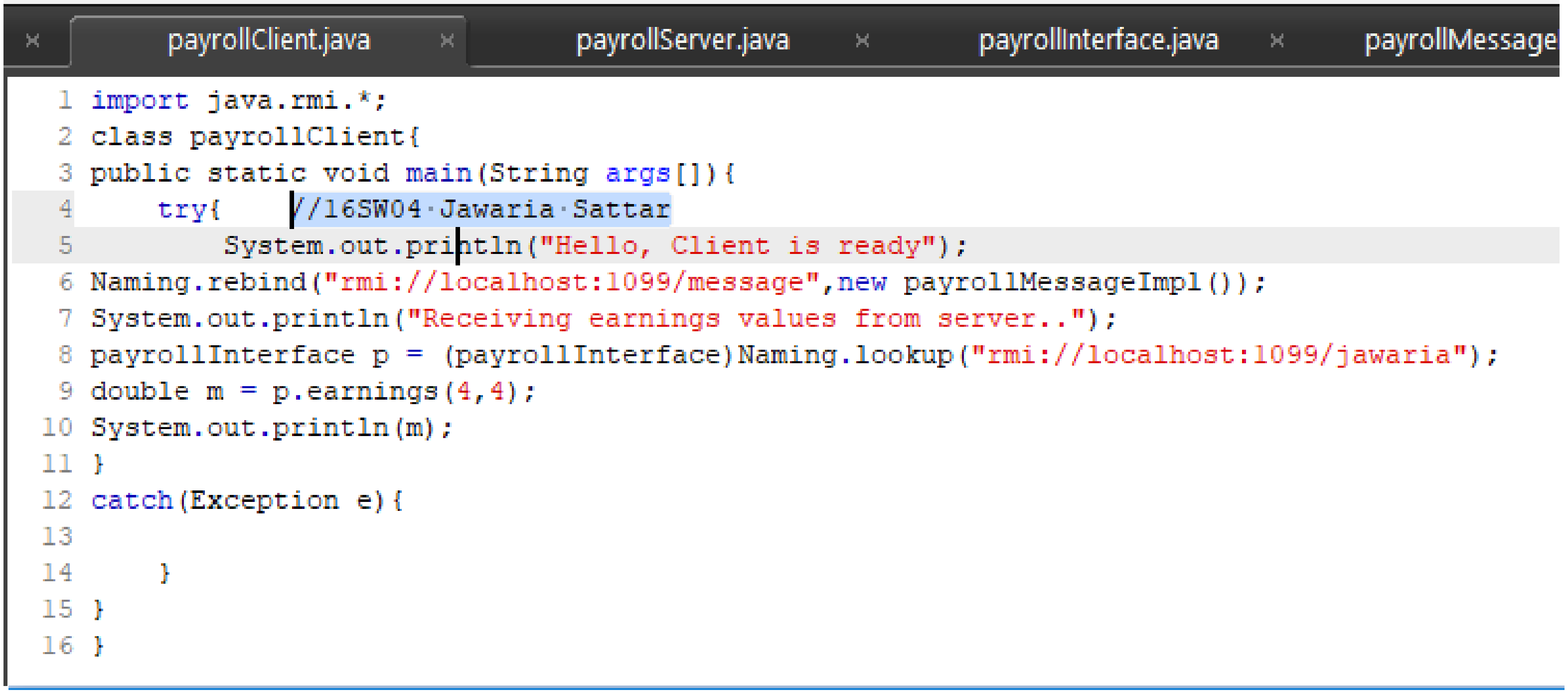
```
import java.rmi.*;
public interface payrollMessageInterface extends Remote{
public void message(String str) throws RemoteException;
}    //16SW04 Jawaria Sattar
```



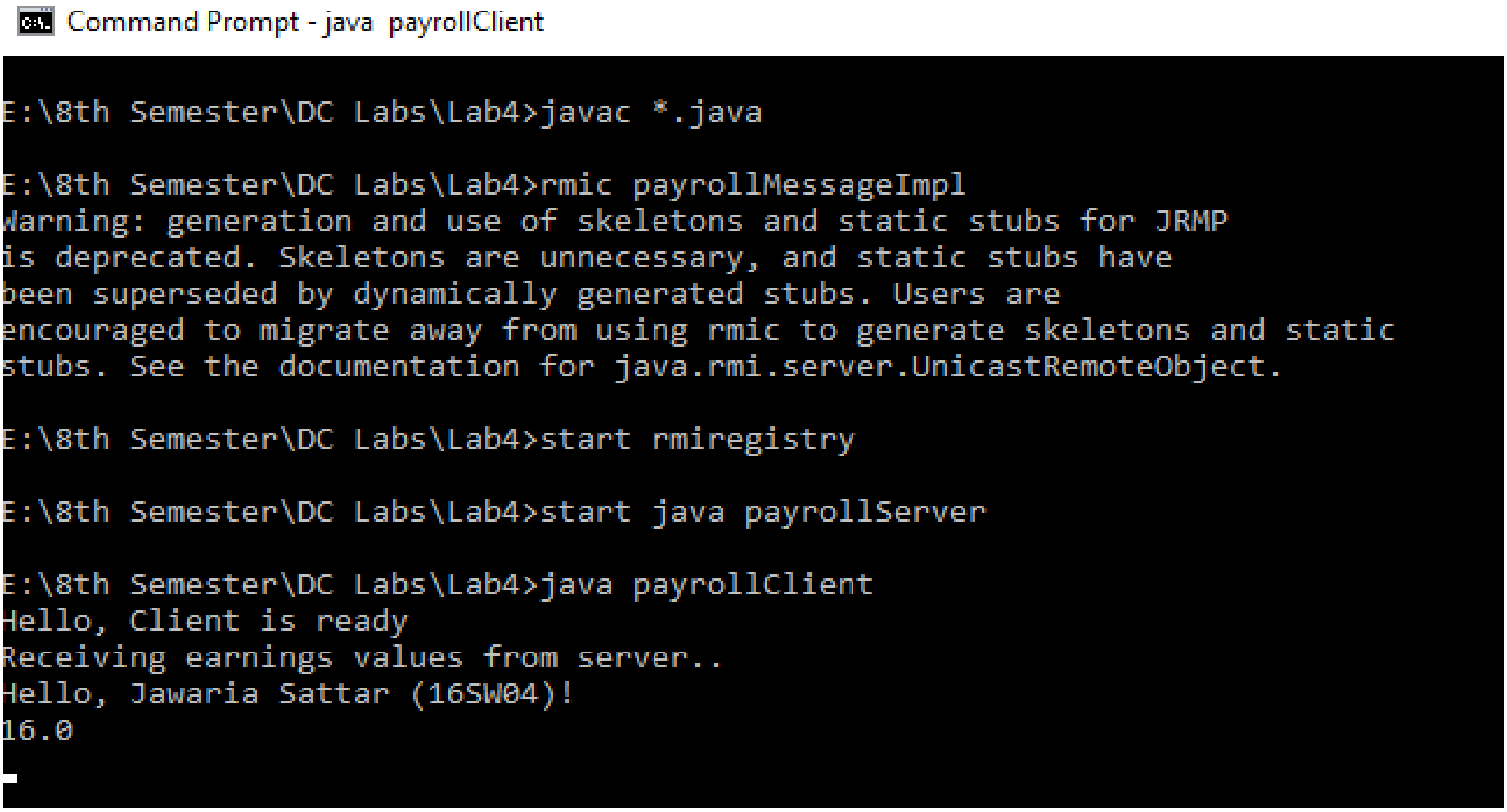
```
import java.rmi.*;
import java.rmi.server.*;
 public class payrollMessageImpl extends UnicastRemoteObject implements payrollMessageInterface{
     private String message;
public payrollMessageImpl() throws RemoteException{

}      //16SW04 Jawaria Sattar
public void message(String str)throws RemoteException{
    System.out.println("Hello, Jawaria Sattar (16SW04)!");
}
}
```



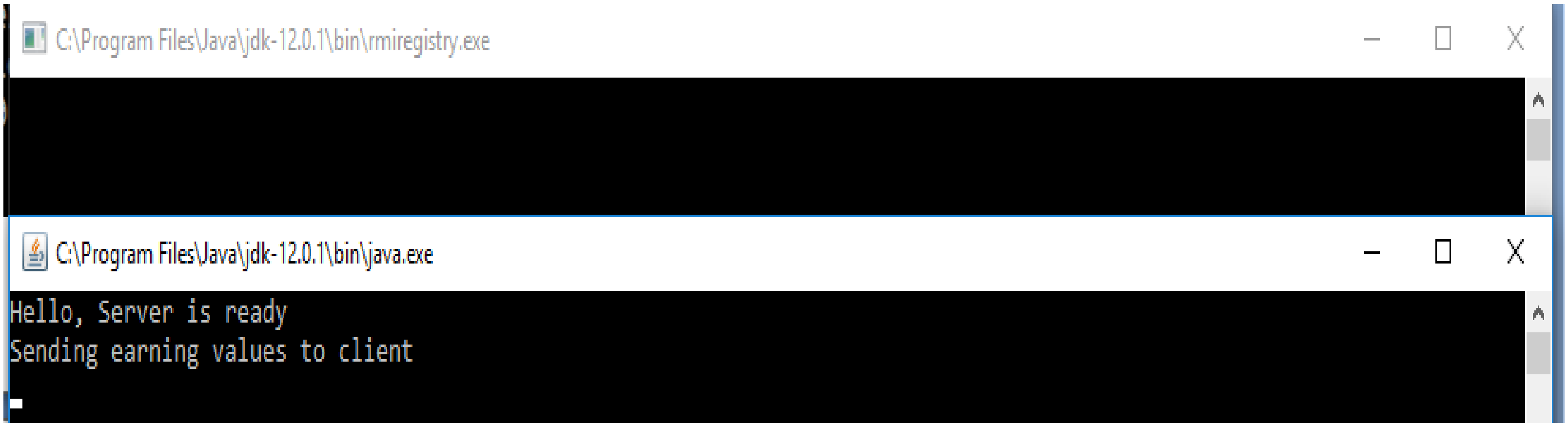
```
import java.rmi.*;
public class payrollServer{
public static void main(String args[]){
     try{   //16SW04 Jawaria Sattar
String ro = "rmi://localhost:1099/jawaria";
payrollImpl p = new payrollImpl();
Naming.rebind(ro,p);
System.out.println("Hello, Server is ready");
System.out.println("Sending earning values to client");}
catch(Exception e){
     }
}
}
```



```java
import java.rmi.*;
class payrollClient{
public static void main(String args[]){
    try{     //16SW04 Jawaria Sattar
        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/jawaria");
double m = p.earnings(4,4);
System.out.println(m);
}
catch(Exception e){

    }
}
}
```



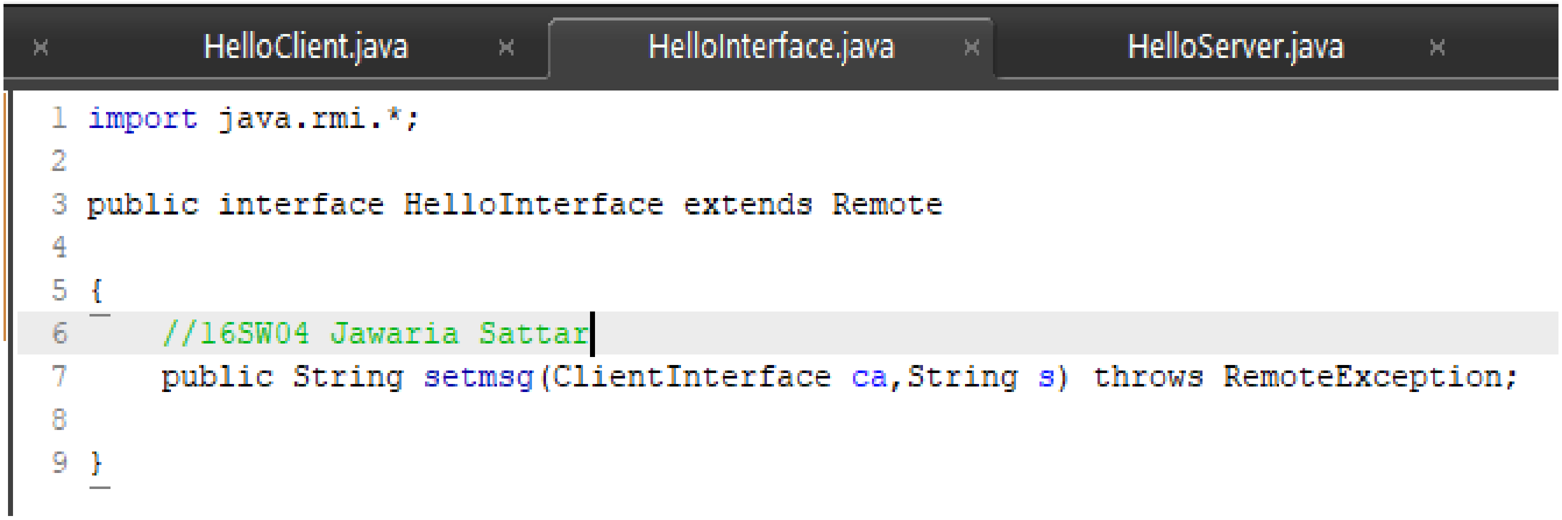
```
E:\8th Semester\DC Labs\Lab4>javac *.java

E:\8th Semester\DC Labs\Lab4>rmic payrollMessageImpl
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.

E:\8th Semester\DC Labs\Lab4>start rmiregistry

E:\8th Semester\DC Labs\Lab4>start java payrollServer

E:\8th Semester\DC Labs\Lab4>java payrollClient
Hello, Client is ready
Receiving earnings values from server..
Hello, Jawaria Sattar (16SW04)!
16.0
```


C:\Program Files\Java\jdk-12.0.1\bin\rmiregistry.exe
C:\Program Files\Java\jdk-12.0.1\bin\java.exe

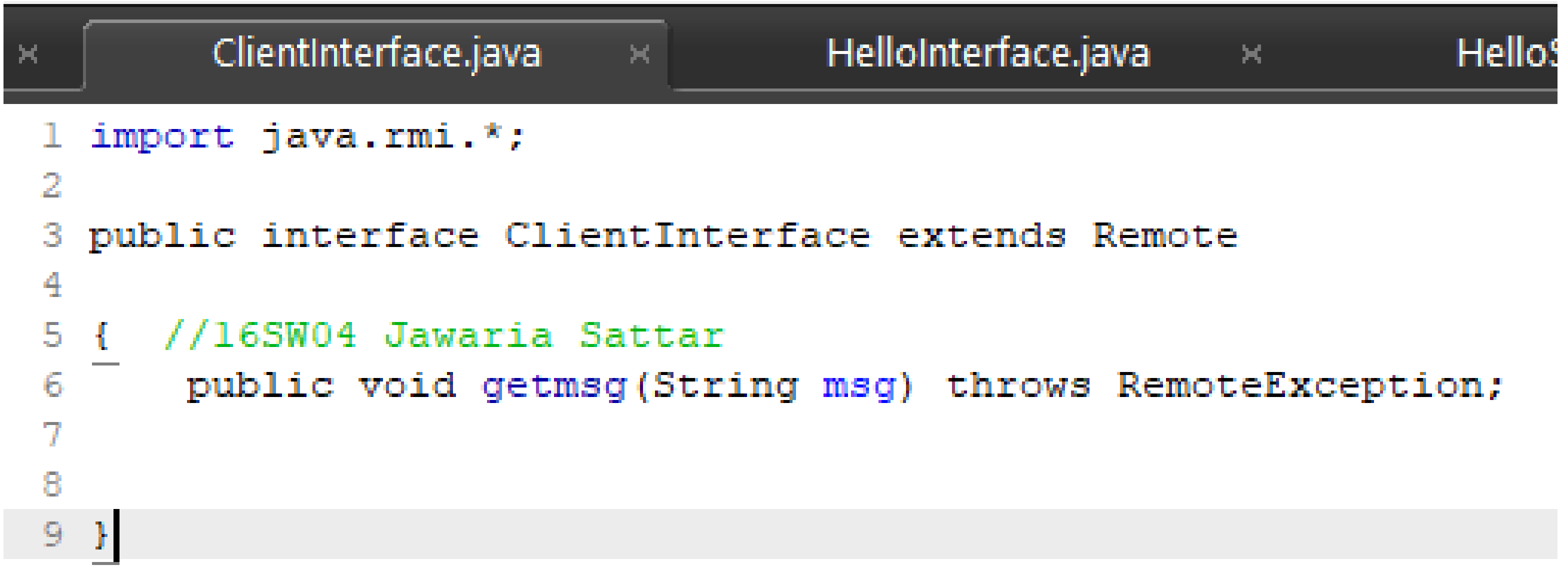
```
Hello, Server is ready
Sending earning values to client
```

2. **Optional Task:**

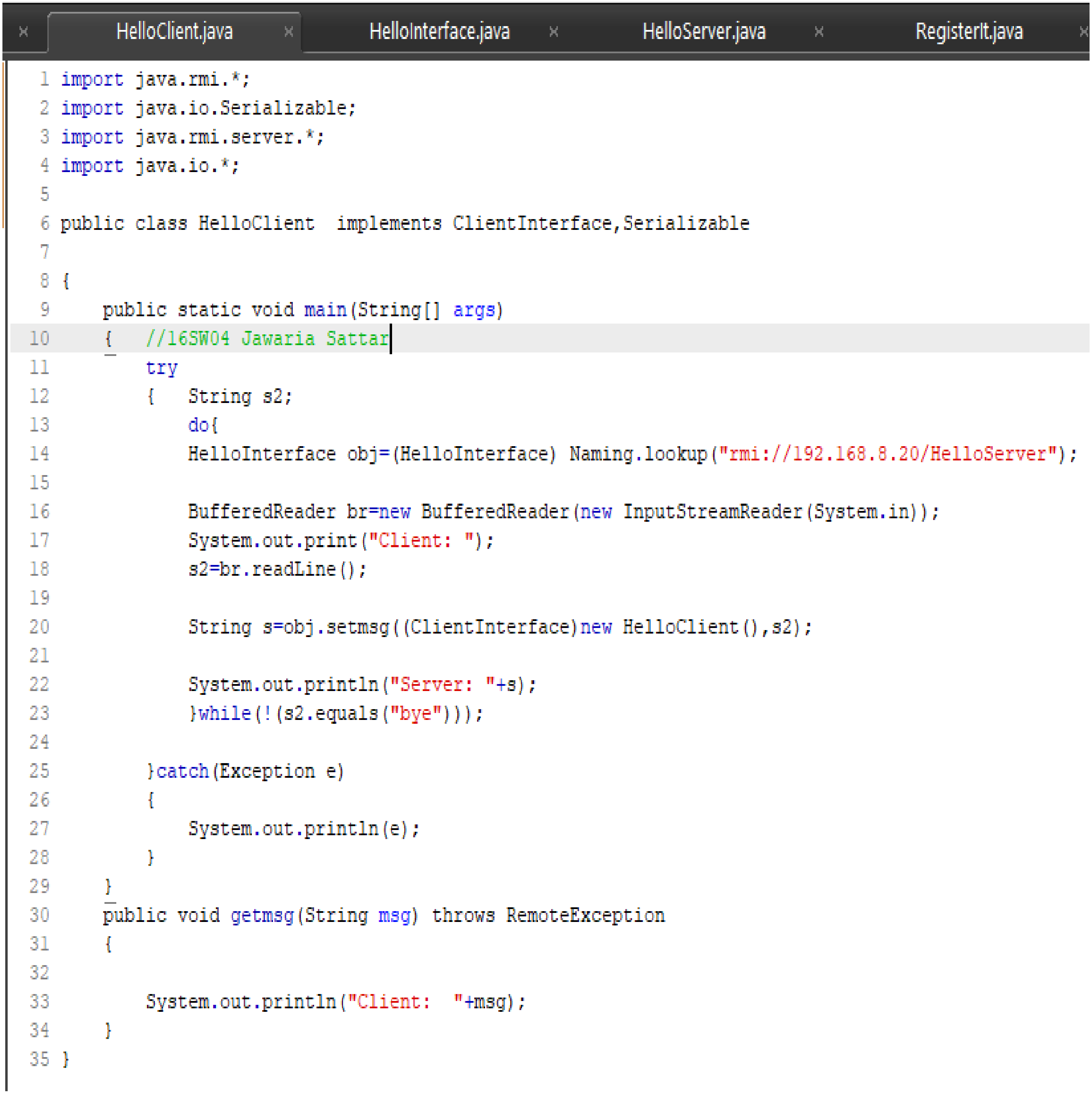
Develop an RMI application for chatting between client and server , the chatting should stop when both of them say “bye”.



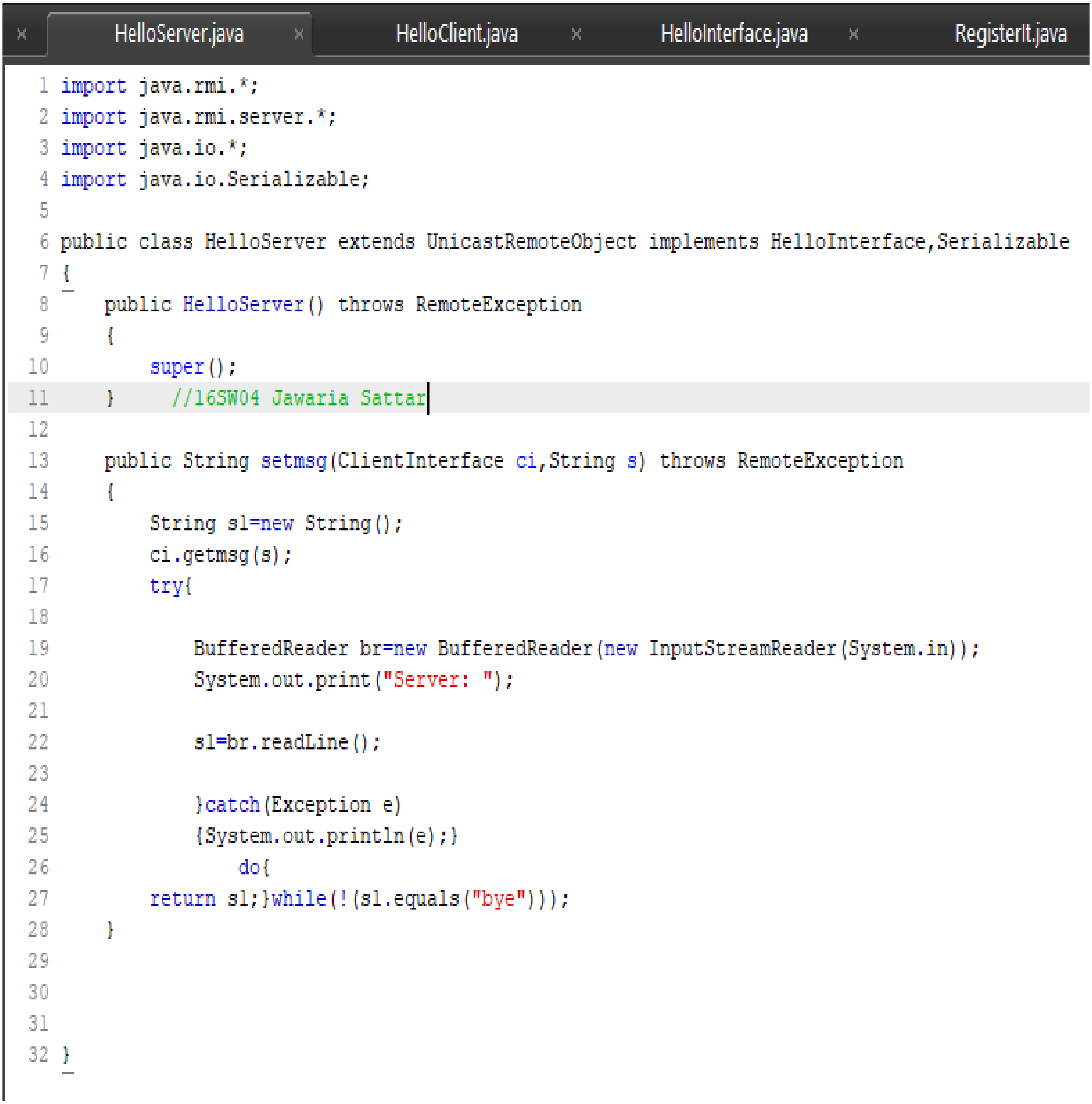
```java
import java.rmi.*;

public interface HelloInterface extends Remote

{
    //16SW04 Jawaria Sattar
    public String setmsg(ClientInterface ca,String s) throws RemoteException;

}
```



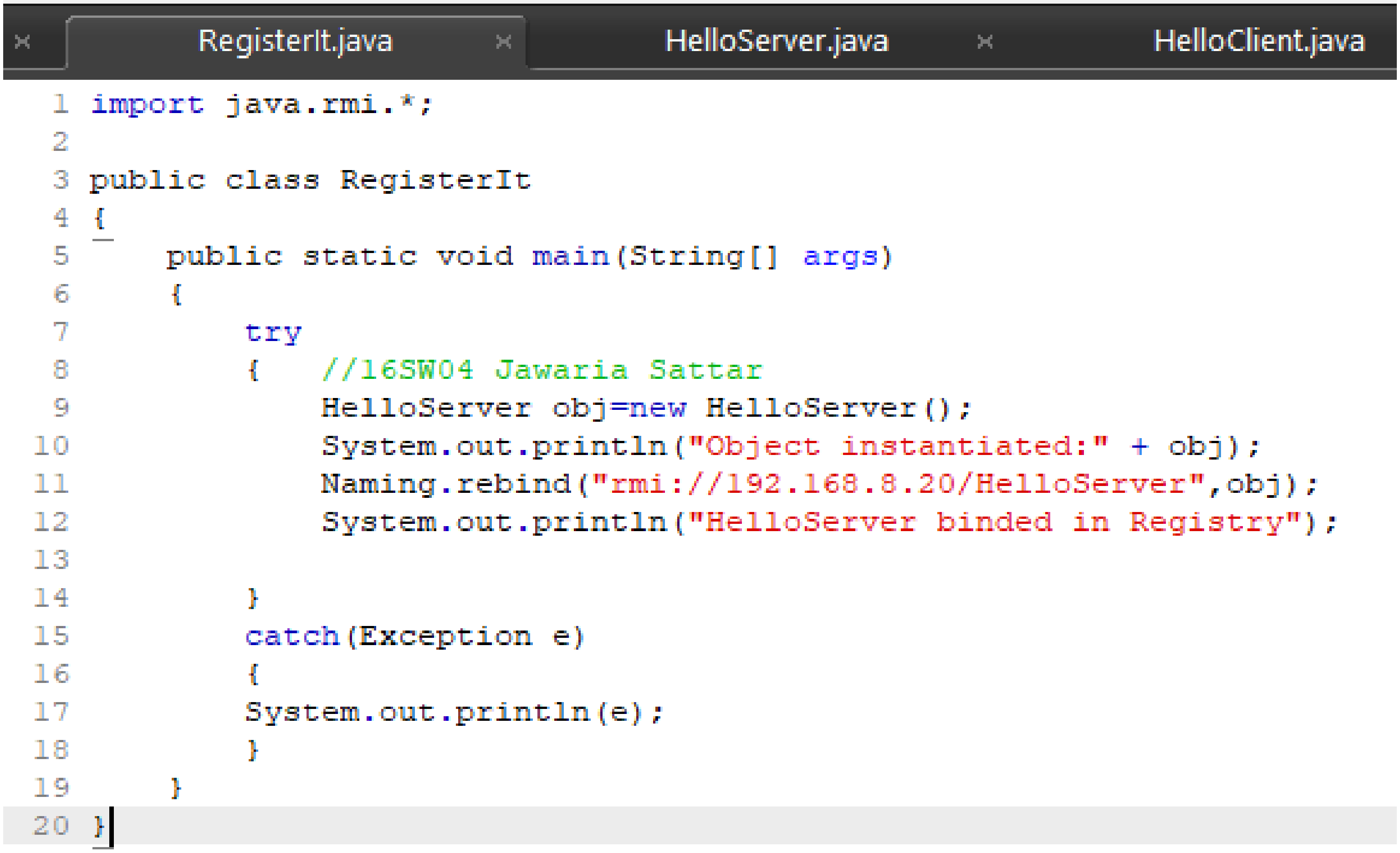
```java
import java.rmi.*;

public interface ClientInterface extends Remote

{   //16SW04 Jawaria Sattar
     public void getmsg(String msg) throws RemoteException;


}
```



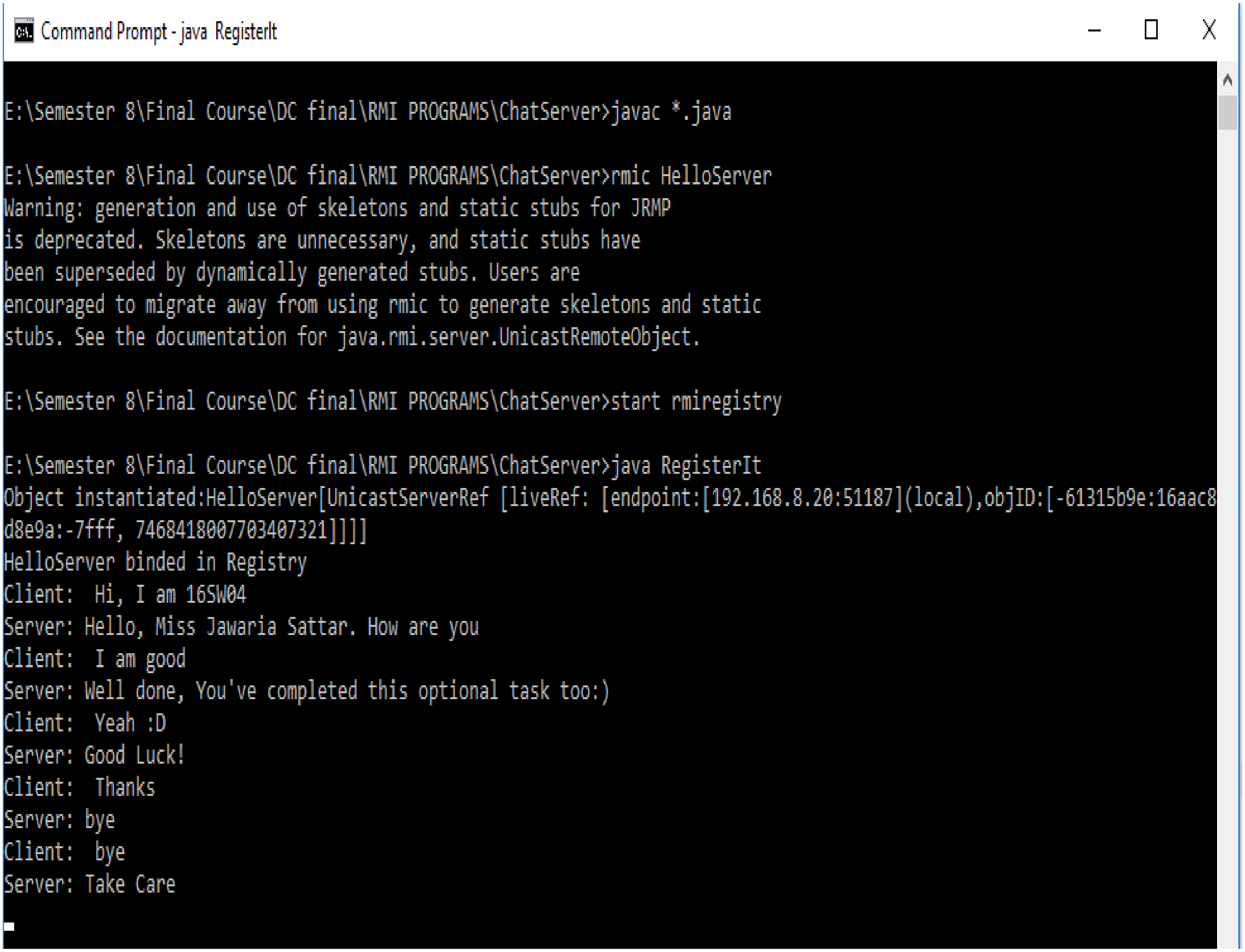
```
import java.rmi.*;
import java.io.Serializable;
import java.rmi.server.*;
import java.io.*;

public class HelloClient  implements ClientInterface,Serializable

{
    public static void main(String[] args)
    {   //16SW04 Jawaria Sattar
        try
        {   String s2;
            do{
            HelloInterface obj=(HelloInterface) Naming.lookup("rmi://192.168.8.20/HelloServer");

            BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
            System.out.print("Client: ");
            s2=br.readLine();

            String s=obj.setmsg((ClientInterface)new HelloClient(),s2);

            System.out.println("Server: "+s);
            }while(!(s2.equals("bye")));

        }catch(Exception e)
        {
            System.out.println(e);
        }
    }
    public void getmsg(String msg) throws RemoteException
    {

        System.out.println("Client:  "+msg);
    }
}
```



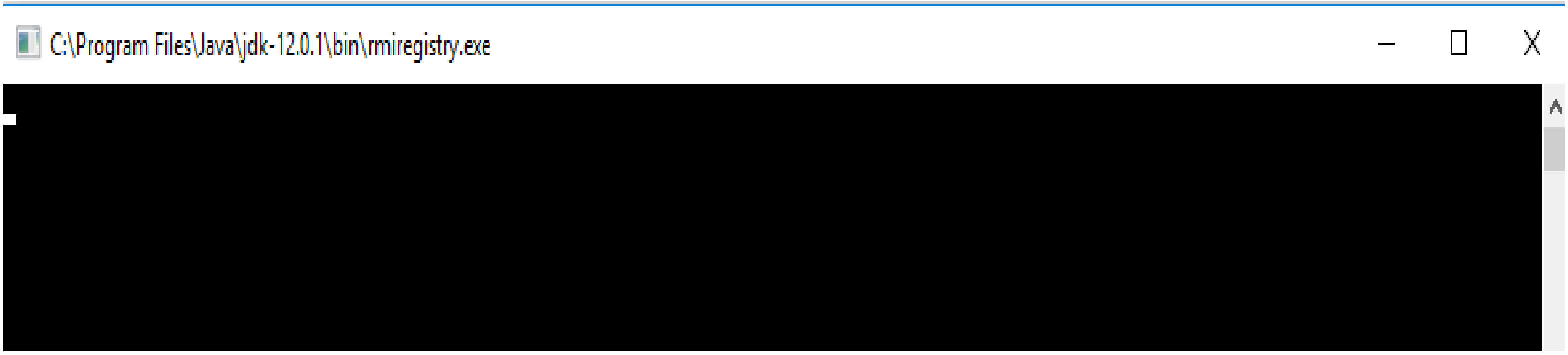
```java
import java.rmi.*;
import java.rmi.server.*;
import java.io.*;
import java.io.Serializable;

public class HelloServer extends UnicastRemoteObject implements HelloInterface,Serializable
{
    public HelloServer() throws RemoteException
    {
        super();
    }     //16SW04 Jawaria Sattar

    public String setmsg(ClientInterface ci,String s) throws RemoteException
    {
        String s1=new String();
        ci.getmsg(s);
        try{

            BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
            System.out.print("Server: ");

            s1=br.readLine();

            }catch(Exception e)
            {System.out.println(e);}
                do{
        return s1;}while(!(s1.equals("bye")));
    }



}
```

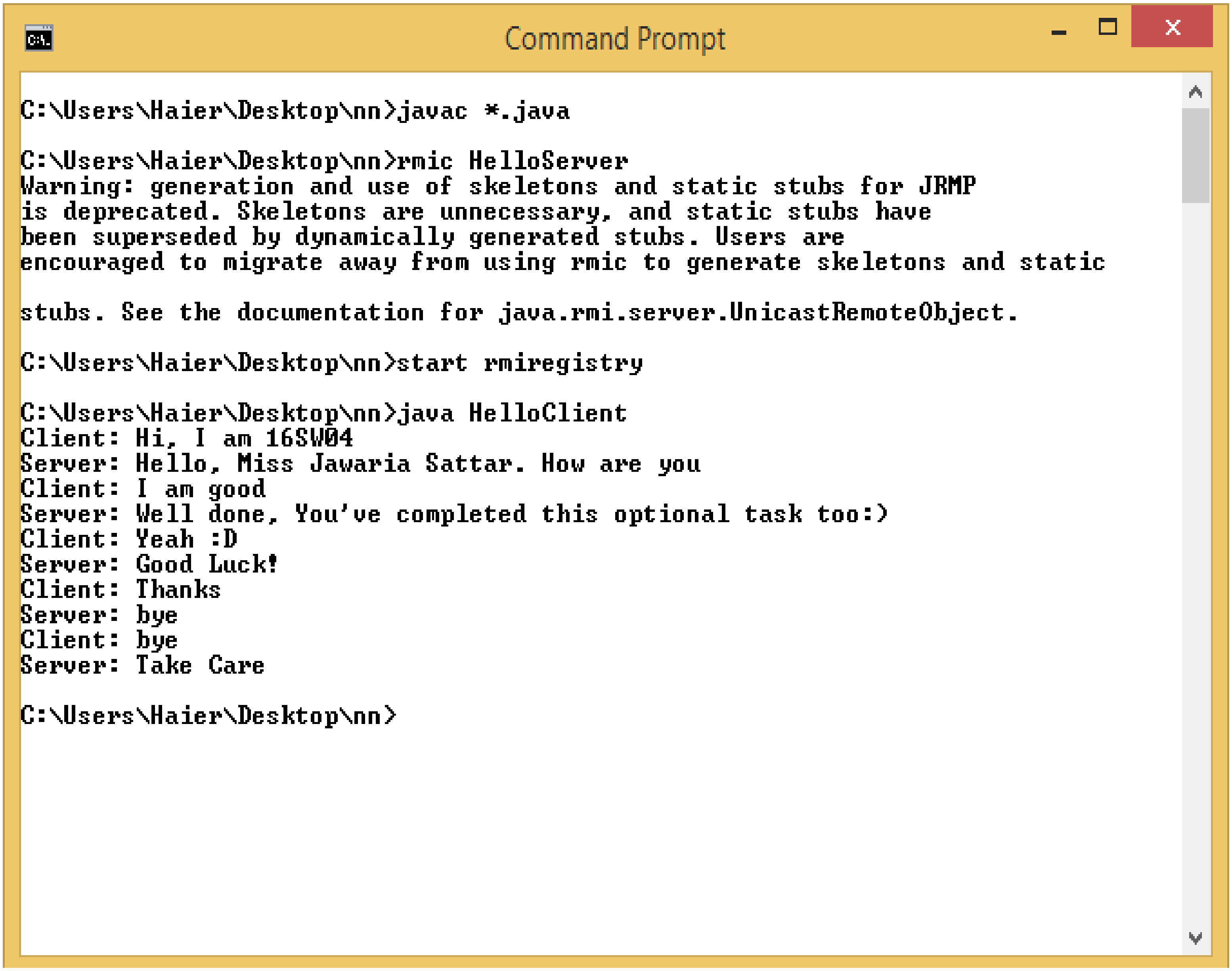


```java
import java.rmi.*;

public class RegisterIt
{
    public static void main(String[] args)
    {
        try
        {   //16SW04 Jawaria Sattar
            HelloServer obj=new HelloServer();
            System.out.println("Object instantiated:" + obj);
            Naming.rebind("rmi://192.168.8.20/HelloServer",obj);
            System.out.println("HelloServer binded in Registry");

        }
        catch(Exception e)
        {
        System.out.println(e);
        }
    }
}
```



```
E:\Semester 8\Final Course\DC final\RMI PROGRAMS\ChatServer>javac *.java

E:\Semester 8\Final Course\DC final\RMI PROGRAMS\ChatServer>rmic HelloServer
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.

E:\Semester 8\Final Course\DC final\RMI PROGRAMS\ChatServer>start rmiregistry

E:\Semester 8\Final Course\DC final\RMI PROGRAMS\ChatServer>java RegisterIt
Object instantiated:HelloServer[UnicastServerRef [liveRef: [endpoint:[192.168.8.20:51187](local),objID:[-61315b9e:16aac8
d8e9a:-7fff, 7468418007703407321]]]]
HelloServer binded in Registry
Client:  Hi, I am 16SW04
Server: Hello, Miss Jawaria Sattar. How are you
Client:  I am good
Server: Well done, You've completed this optional task too:)
Client:  Yeah :D
Server: Good Luck!
Client:  Thanks
Server: bye
Client:  bye
Server: Take Care
```





```
C:\Users\Haier\Desktop\nn>javac *.java

C:\Users\Haier\Desktop\nn>rmic HelloServer
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.

C:\Users\Haier\Desktop\nn>start rmiregistry

C:\Users\Haier\Desktop\nn>java HelloClient
Client: Hi, I am 16SW04
Server: Hello, Miss Jawaria Sattar. How are you
Client: I am good
Server: Well done, You've completed this optional task too:)
Client: Yeah :D
Server: Good Luck!
Client: Thanks
Server: bye
Client: bye
Server: Take Care

C:\Users\Haier\Desktop\nn>
```