//Search.java

//Creating a search interface

import java.rmi.\*;

public interface Search extends Remote

{

//Declaring the method prototype

public String query(String search) throws

RemoteException;

}

//SearchQuery.java

import java.rmi.\*;

import java.rmi.server.\*;

public class SearchQuery extends RemoteObject

implements Search

{

//Implementation of query interface

public String query(String search)

throws RemoteException{

String result;

if(search.equals("Reflection in java"))

result="Found";

else

result="Not Found";

return result;

}

}

//Server.java

import java.net.\*;

import java.io.\*;

public class Server

{

private Socket

socket=null;

private ServerSocket server =null;

private DataInputStream in =null;

//constructor with port

public Server(int port)

{

//start server and wait for connection

try

{

server= new ServerSocket(port);

System.out.println("Sevrer started");

System.out.println("Waiting for client");socket=server.accept();

System.out.println("Cliebt accepeted");

// i/p from clinet

in=new DataInputStream(new BufferedInputStream(socket.getInputStream()));

String line="";

//read msg from client until'over' is sent

while(!line.equals("Over"))

{

try

{

line=in.readUTF();

System.out.println(line);

}

catch(IOException i)

{

System.out.println(i);

}

}

System.out.println("Closing Connnection");

//close connection

socket.close();

in.close();

}

catch(IOException i)

{

System.out.println(i);

}

}

public static void main(String args[])

{

Server server =new Server(5000);

}

}

//Client.java

import java.io.\*;

import java.net.\*;

public class Client{

private Socket socket = null;

private BufferedReader d = null;

private InputStream input = null;

private DataOutputStream out = null;

public Client(String address, int port)

{

try{

socket = new Socket(address, port);

System.out.println("Connected");

System.out.println("Done with 1st program of DS");

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

out = new DataOutputStream(

socket.getOutputStream());

}

catch(UnknownHostException u)

{

System.out.println(u);

return;

}

catch(IOException i )

{

System.out.println(i);

return;

}

String line = "";

while(!line.equals("Over"))

{

try{

line = d.readLine();

out.writeUTF(line);

}

catch(IOException i )

{

System.out.println(i);

}

}

try{

input.close();

out.close();

socket.close();

}

catch(IOException i)

{

System.out.println(i);

}

}

public static void main(String args[])

{

Client client = new Client("127.0.0.1", 5000);

}

}