package client;

import java.io.DataInputStream; //reads information from the stream

import java.io.DataOutputStream; //enables software to take use of the input stream for reading simple data

import java.io.IOException; //When an input or output operation fails or is misinterpreted,

// an input/output exception (I/O) is thrown.

import java.net.Socket; //The ServerSocket Class is for implementing the server's end of a

// client/server Socket Connection in a platform-independent manner.

import java.net.UnknownHostException;//Since each instance of the socket class is uniquely

// bound to a single distant host, establishing a connection to a different host requires the creation of a new socket object.

//Perform the necessary setup on the input stream as well as socket.

public class Client {

//Set the output and input streams just on socket to their default values.

private Socket socket = null;//assigns the null vale to the socket

private DataInputStream input = null; //data input stream is assigned a null value

private DataOutputStream out = null; // the same proceed is repeated to the same

// constructor for entering the port number and the ip address

public Client(String address, int port) {

try { //it will utilize available address and port

socket = new Socket(address, port);// will create a new socket and address

System.out.println("connected");// when the connection is established his message will be displayed

System.out.println("hello world"); // also this will be displayed

input = new DataInputStream(System.in);

// sends output to the socket

out = new DataOutputStream(socket.getOutputStream()); // this will send output from the client

} catch (UnknownHostException u) { //wil try to connect with available host server

System.out.println(u); // it will display the connection message after conncetion

} catch (IOException i) { // tries to catch errors if may arise

System.out.println(i); //prits the error catched

}// message that will be read from the input string

String line = "";

// continue reading until the word "Over" is entered.

//and until the connection is disconnected

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

try {

line = input.readLine();

out.writeUTF(line); // use the enconding method

} catch (IOException i) { // tries to catch any error if they may occur

System.out.println(i); // prints the errors

}

}

// pull the plug on the connection.

// after the connection is broken or if the client decides to close

try {

input.close(); //closes the input

out.close(); //closes the output

socket.close(); //closes the socket

} catch (IOException i) {

System.out.println(i);

}

}

public static void main(String args[]) {

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

}

}

Task 2

Set the output and input streams just on socket to their default values.