

# PROGRAMMING TECHNOLOGIES AND EDUCATIONAL PRACTICE



## PROJECT - 3

### Client - Server Communication

---

In your Final Project you will create a program where a client and a server establishes a simple communication between them. One of the key packages in the Java system is the “java.net library”. This provides support for accessing the Internet both as a client accessing remote services and as a server providing services to remote clients.

#### **Port:**

A computer connected to the Internet via the Internet protocol TCP/IP has a set of port numbers attached to it that identify the application that is to be run. The port numbers 0 – 1024 are used for frequently used processes and applications.

#### **Socket:**

All communication by computer on the Internet is carried out by means of sockets. A socket is made up of two entities: a port and an IP address. The combination of socket and port is unique and hence can be used by Internet software to determine the communication circuit along with an application such as email. In this way computers on the Internet can be easily connected together.

### **Introduction**

#### **Part - 1**

This is a client/server implementation for transmitting packets. Each packet consists of two parts; the header part containing only the “**serial number**” and the “**data part**” containing a message (any string of characters). Client sends a packet to the server and receives its acknowledgment, then enters a text and server converts it into uppercase letters and returns it back. When a client sends a “**CLOSE**” message, both client and server terminate.

Compile all three files. Run server and client files in two different windows.

#### **1. TCPPacketServer.java**

- Use /\*\* The getOutputStream() method to send out the data through the socket stream \*\*/
- Use /\*\* The getInputStream() method to obtain a handle to the socket stream \*\*/

#### **2. TCPPacketClient.java**

- Use `/**` the instance of Socket class uses the address of this computer and the port number 1024 `*/`
- To learn your host name use commands below:
  - a. `> ipconfig /all (Windows + R)`
- Use `/**` The `getInputStream()` method to obtain a handle to the socket stream `*/`
- Use `/**` The `getOutputStream()` method to send out the data through the socket stream `*/`
- Use `BufferedReader` method `/**` to read from the keyboard the data (Packet) that needs to be send to the server `*/`
- `/**` Create a new instance of Packet class with data's header,the data (packet) and packet number `*/`

### 3. Packet.java

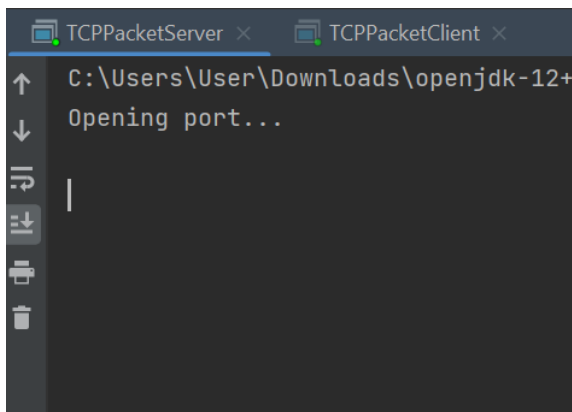
- `private int serialNo;`
  - `private String data;`
- ```

public String getData() {
    return data;
}

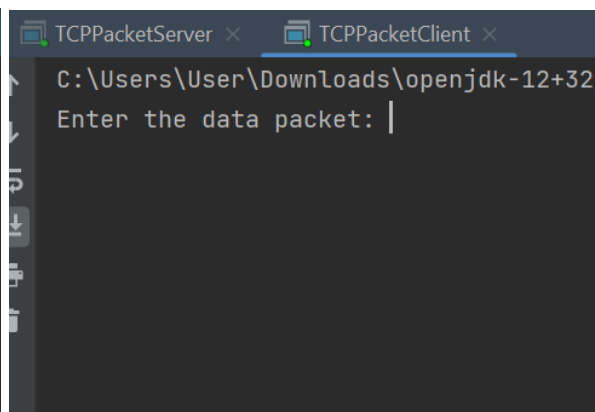
public int getSerialNo() {
    return serialNo;
}

```

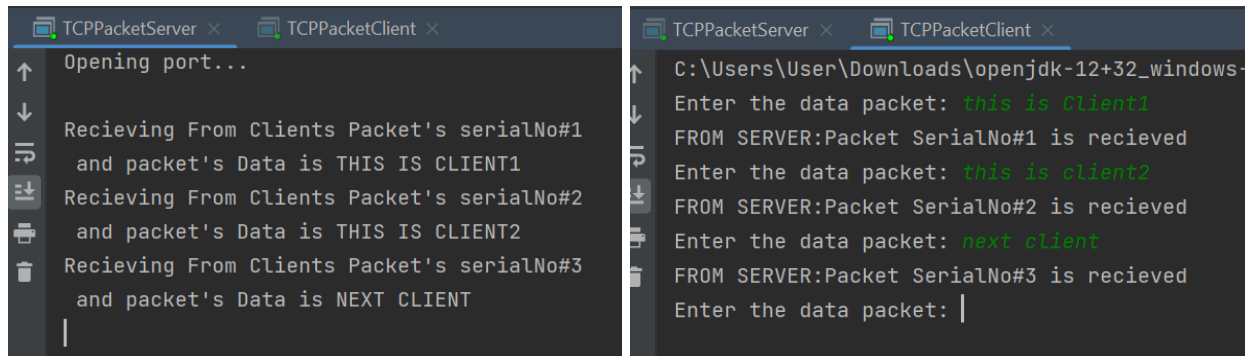
#### Step - 1: Compile TCPPacketServer



#### Step - 2: Compile TCPPacketClient



### Step - 3: Enter commands to the data packet:



The image shows two terminal windows side-by-side. The left window is titled 'TCPpacketServer' and the right is 'TCPpacketClient'. The server window shows the process of opening a port and receiving three packets from the client, each with a serial number and data. The client window shows the user entering the data for each packet, which is then received by the server.

```
TCPpacketServer × TCPpacketClient ×
Opening port...
Receiving From Clients Packet's serialNo#1
and packet's Data is THIS IS CLIENT1
Receiving From Clients Packet's serialNo#2
and packet's Data is THIS IS CLIENT2
Receiving From Clients Packet's serialNo#3
and packet's Data is NEXT CLIENT

C:\Users\User\Downloads\openjdk-12+32_windows-
Enter the data packet: this is client1
FROM SERVER:Packet SerialNo#1 is recieved
Enter the data packet: this is client2
FROM SERVER:Packet SerialNo#2 is recieved
Enter the data packet: next client
FROM SERVER:Packet SerialNo#3 is recieved
Enter the data packet: |
```

## Part - 2

In this part you will create a Client and Server connection.

A client application generates a socket on its end of the connection and attempts to connect that socket to a server. When the connection is established, the server constructs a socket object on its end of the communication. The client and server can now communicate by writing to and reading from the socket using the **java.net** package.

Use(**Mandatory**):

- `PrintStream ps = new PrintStream(socket.getOutputStream());` to send a message to Client from Server.

Use(**Mandatory**):

- `BufferedReader br = new BufferedReader(new InputStreamReader(socket.getInputStream()));` to receive the message from Client.

### Step - 1: Compile Serve.java class

```
MTServer x Clients x
C:\Users\User\Downloads\openjdk-12+32_w...
Waiting for client request
New client is pop up!
Server:
```

### Step - 2: Compile Client.java class

```
MTServer x Clients x
C:\Users\User\Downloads\openjdk-12+32_w...
Request sent successfully
|
```

### Step - 3: Client and Server connection

```
MTServer x Clients x
C:\Users\User\Downloads\openjdk-12+32_windo...
Waiting for client request
New client is pop up!
Server: Hella
Clients: Hi
Server: How are you ?
Clients: I am fine
Server: Welcome to my app
Clients: Nice to see you
Server: |
```

```
MTServer x Clients x
C:\Users\User\Downloads\openjdk-12+32_windo...
Request sent successfully
Server: Hello
Client: H1
Server: How are you ?
Client: I am fine
Server: Welcome to my app
Client: Nice to see you
|
```

## UML DIAGRAM



## **Notes on Grading**

Make sure that every class does exactly what is described in the project requirements. Do not attempt to show some ready implementation downloaded from the Internet. It shall be considered as plagiarism, and get ZERO points. A fully tested, compiling, and well-documented class will be worth some points even if you do not finish the whole assignment. Any kind of extra improvements on the mechanism is welcome and may be worth some bonus points (up to +10 points).

Do NOT share your code with anyone. It would be considered academic dishonesty, and strictly penalized. All the works submitted shall be inspected by a special program and reviewed by the instructors. Any kind of similarity, or not being able to answer questions on the project gives the instructor full right to penalize the work, and even cancel the results that have already been graded because of cheating issues. Simply explained, the fact that you did not cheat yourself, or that you showed your work and it has already been graded is not enough. All the sides included in cheating (which is a crime) will be penalized. Again, do not share your code under any conditions.