

# COMP 6231: Distributed Systems Design

## Lab Instructions

Using Java Datagrams

May 15, 2022

### 1 General Information

**Lab Date:** Tuesday, May 17<sup>th</sup>, 2022.

Your lab instructor will provide you with instructions on how to do this lab activity.

### 2 Overview

The purpose of this lab is to get a hands-on experience on using datagrams. In this lab you expected to develop a UDP a listener that adds two integers and sends the result back to its peers.

As a starting point you are given a simple EchoServer and EchoClient. They work as follows:

- The echo client sends messages to the echo server. It exits if message is "exit".
- The echo server replies back (aka echoes) to the client the same message it received.

### 3 Instructions

You may start from scratch or use the provided project template.

Clone the template from the second tutorial on Github:

<https://github.com/COMP6231/T02>

1. Compile and run EchoServer on any port you like.
2. Compile and run EchoClients on *localhost* binding to the server's port.
3. Using client send the first integer to the server.
4. The server replies to the client with message: **"send me 2nd term"**.
5. Using client send the second integer to the server.
6. The server replies with the sum.

Repeat the same with two clients. Make sure the sum is correctly calculated for each client even if sent integers are not sent in sequence. For example:

1. Client-1 sends 1st integer to the server.
2. The server replies to the client-1 with message: **"send me 2nd term"**.
3. Client-2 sends 1st integer to the server.
4. The server replies to the client-2 with message: **"send me 2nd term"**.
5. Client-1 sends 2nd integer to the server.
6. The server replies to the client-1 with the sum.
7. Client-2 sends 2nd integer to the server.
8. The server replies to client-2 with the sum.

### 4 After the Lab

Suppose the server is not running on the local host and clients don't know the server's IP address and/or how many echo servers are running on the network. What would be a way to discover such servers? Hint: Think of a discovery protocol.

# References

1. A Guide To UDP In Java:

<https://www.baeldung.com/udp-in-java>

2. Echo Server using Java:

<https://www.demo2s.com/java/java-creating-a-udp-echo-server.html>

3. Echo Client using Java:

<https://www.demo2s.com/java/java-creating-a-udp-echo-client.html>

4. **DatagramSocket.setReuseAddress()**:

<https://docs.oracle.com/javase/7/docs/api/java/net/DatagramSocket.html>