

# CSCE 416 Programming Assignment 1

## Getting Started on Network Programming

The purpose of this assignment is to familiarize you with network programming. You will obtain, compile, and run a simple network program. You will then implement an extension.

Download the file `OneWayMesg.zip` from the [Blackboard](#) and unzip it. You will find three folders, 'c', 'java', and 'python', corresponding to an implementation of the one-way messaging application in that programming language. Each folder contains two programs, `OneWayMesgServer` and `OneWayMesgClient`. The instructions on how to run them are given in the [README.txt](#) file in each folder.

To see the interaction, run `OneWayMesgClient` and `OneWayMesgServer` in two separate terminals. Type something on the terminal where `OneWayMesgClient` is running, and see that `OneWayMesgServer` prints out the message. Note that `OneWayMesgServer` and `OneWayMesgClient` use a specific port to communicate.

Your task is to modify `OneWayMesgServer` to create `TwoWayMesgServer` that accepts user input and sends it to the `TwoWayMesgClient`, which is a modified version of `OneWayMesgClient` that prints the received message on the terminal. You can complete this assignment in 'c/c++', 'java', or 'python' language. Your server and client program should run with the following (python as an example):

```
Server> python TwoWayMesgServer.py portNumber serverName
Client> python TwoWayMesgClient.py localhost portNumber clientName
```

The `TwoWayMesgServer` and `TwoWayMesgClient` output should look like this:

```
Server> Waiting for a client ...
Server> Connected to a client at ('127.0.0.1', '60681')
Server> clientName: Hello Server
Server> Hello Client
-
Client> Connected to server at ('localhost', '50000')
Client> Hello Server
Client> serverName: Hello Client
```

That's it. Sounds simple, doesn't it? Indeed, for experienced programmers, this assignment is trivial. Others should find this as an easy way to get started on network programming. Don't simply download the source code and compile the programs; but make sure that you understand how the sockets are created and the connection is established. Start early and seek as much help as needed from the TA and the instructor.

### Submission

- (1) Create a zip file containing the new `TwoWayMesgServer` and `TwoWayMesgClient` source files
- (2) Rename the zip file as `YOURLASTNAME_P1.zip` (YOURLASTNAME in all caps)
- (3) Upload it in the [Blackboard](#)

### Grade Breakdown

- (1) 25%: You submitted your assignment on-time; and the new, modified files compile correctly. **Zero points if the files do not compile or you upload the original files.**
- (2) 75%: Both `TwoWayMesgClient` and `TwoWayMesgServer` can print each other's messages.