COSC 4301 – MODERN PROGRAMMING
Project 3 – Client/Server Program (Multi-Threaded Server)

Write a client/server program that reads two integers from the user on the client side separated by a comma and/or spaces. These integers must be entered as one string and let the server parse it out to retrieve the two integers. The server should be **multi-threaded** and should be able to serve multiple clients simultaneously. The server should send back the prime numbers, their sum, mean, and standard deviation back to the client.

The Server must be Thread Safe.

Use the **isPrime** method provided in the file **PrimeTest.java** and have your server listen on port **4301**, the course number.

The server should validate the input from the client as follows and sends an appropriate message back to the client:

1. The data entered are integers.
2. The first integer must be less than the second.
3. Both integers must be greater than zero.

Allow the user to run the program as many times as possible until he/she enters "Bye" to indicate no more data. The client should terminate, and the server should go back to listen for new client connections.

The server code and the client code must be in separate classes. **No input, processing, or output should happen in the main methods.** All work should be delegated to other non-static methods.

All classes in this project must be public, non-static and not nested in other classes.

Every method in your program should be limited to performing a single, well-defined task, and the name of the method should express that task effectively.

Test your program with multiple clients connected to make sure it works correctly and then copy and paste the output to a file named **Program3-output.txt**. Create a folder named, **<YourLastNameFirstName>_Program3**. Copy your source codes and the output file to the folder. **Zip the folder, as a ".zip" file, and upload it to Blackboard**.

Before you upload your project to Blackboard:

- Ensure that your code conforms to the style expectations set out in class and briefly discussed below.

- Make sure your variable names and methods are descriptive and follow standard capitalization conventions.

- Put comments wherever necessary. Comments at the top of each module should include your name, file name, and a description of the module. Comments at the beginning of methods describe what the method does, what the parameters are, and what the return value is. Use comments elsewhere to help your reader follow the flow of your code. **See the Program1.java file for more details.**

- *Program readability and elegance are as important as correctness.* After you have written your method, read and re-read it to eliminate any redundant lines of code, and to make sure variables and methods names are intuitive and relevant.

Read the assignment very carefully to ensure that you have followed all instructions and satisfied all requirements. **You will not get full credit for this project if it is not written as instructed even if it works as expected.**