CIS 212 Assignment 8: 100 (regular) points + 100 extra points

Note: The regular 100 points can be partial, as usual; the extra 100 credits are atomic (i.e., either “0” or “100”), and cannot be partial. In order to have your work on the extra points taken into account, you need to work on the regular points first; only working on the extra part will lead to the 0 grade for this entire assignment.

The goal of this assignment is to gain experience with network programming, specifically with sockets in Java. This assignment will also require you to develop a server and a client simultaneously. The client will allow the user to enter an arbitrary list of integers to be sent to the server. The server will then return a list of prime integers contained in the original list.

1. [10] Create two classes, one for your client and one for your server. Each class will need a main method so that both classes can be executed.

2. [40] Implement your client to prompt the user to enter a list of integers (and use “!” to end the list.) Send the list of integers to the server (hint: see java.net.Socket), print the list of integers sent to the server, and later print the list of prime integers received from the server.

3. [40] Implement your server to wait for a socket connection (hint: see java.net.ServerSocket). When a socket is accepted, read a list of integers, determine which in the list are prime, and then send a list containing the prime integers back to the client. Your server may or may not shut down after completing this process.

Your client output should look something like:

```
Enter an integer (“!” to send):

3

Enter an integer (“!” to send):

4

Enter an integer (“!” to send):

5

Enter an integer (“!” to send):

7

Enter an integer (“!” to send):

!

Send: [3, 4, 5, 7]

Receive: [3, 5, 7]
```

4. [10] Write code that is clear and efficient. Specifically, your code should be indented with respect to code blocks, avoid unnecessarily repetitive code, avoid code that is commented out or otherwise unused, use descriptive and consistent class/method/variable names, etc.

5. [+100] (Extra credits)

- Write a new client which generates a list of 5 integers randomly in the range [2, 100), prints them, sends them to the server, and receives and prints the list of prime integers contained in the original list. Your client sleeps for 1 second every time after it receives and prints the prime integers. Your client keeps repeating this procedure until stopped by inputting “!”, as specified below. Note in order to keep the server responsive, the server should not shut down after it sends the prime integers back to the client.
- Use “!” to start and stop sending integers to the server, and use “#” to quit the client. Your client needs to be able to read your inputs while sending or receiving data. For example, when you start to run your client, it does nothing; then, when you input “!”, it starts to generate integers, print them, send them, receive and print the results; then, when you input “!” again, it stops (but the program does not quit); then, when you input “!” once again, it resumes to generate integers, print them, send them, receive and print the results again; ... eventually, when you input “#”, the client shuts down and quits. After the client quits, the server may shut down.

Your input/output should look something like:

```
Enter “!” to start and stop, “#” to quit:

!

Send: [12, 18, 32, 50, 82]

Receive: []

Send: [3, 62, 15, 100, 71]

Receive: [3, 71]

!

Enter “!” to start and stop, “#” to quit:

!

Send: [18, 24, 5, 99, 20]

Receive: [5]

Send: [42, 12, 32, 59, 97]

Receive: [59, 97]
```

```
Send: [2, 9, 41, 100, 10]

Receive: [41]

!

Enter “!” to start and stop, “#” to quit:

!

Send: [21, 92, 40, 7, 10]

Receive: [7]

#
```

Note: If you choose not to work on the extra points, please use one folder, put your Java source file(s), i.e., .java file(s), into that folder, rename that folder as [YourName]Assignment8Regular, e.g., BillGatesAssignment8Regular, and then zip the folder into a single zipped file. If you choose to work on the extra points as well, in addition to the folder mentioned above, please have another folder, put your Java source file(s), i.e., .java file(s), on the extra points into that folder, rename that folder as [YourName]Assignment8Extra, e.g., BillGatesAssignment8Extra, and then select both folders and zip both of them altogether into a single zipped file. Finally, you rename your zipped file as [YourName]Assignment8.zip, e.g., BillGatesAssignment8.zip, and then upload it to Canvas. Please do not use the “package” keyword in any of your source file.