John Hodson
UFID 5244-0415
jhodson@ufl.edu

My code may be compiled and run under any environment in which the JDK and the javac compiler is available.

My code is organized as a server and a client. I designed a class for each, with future extensibility in mind.

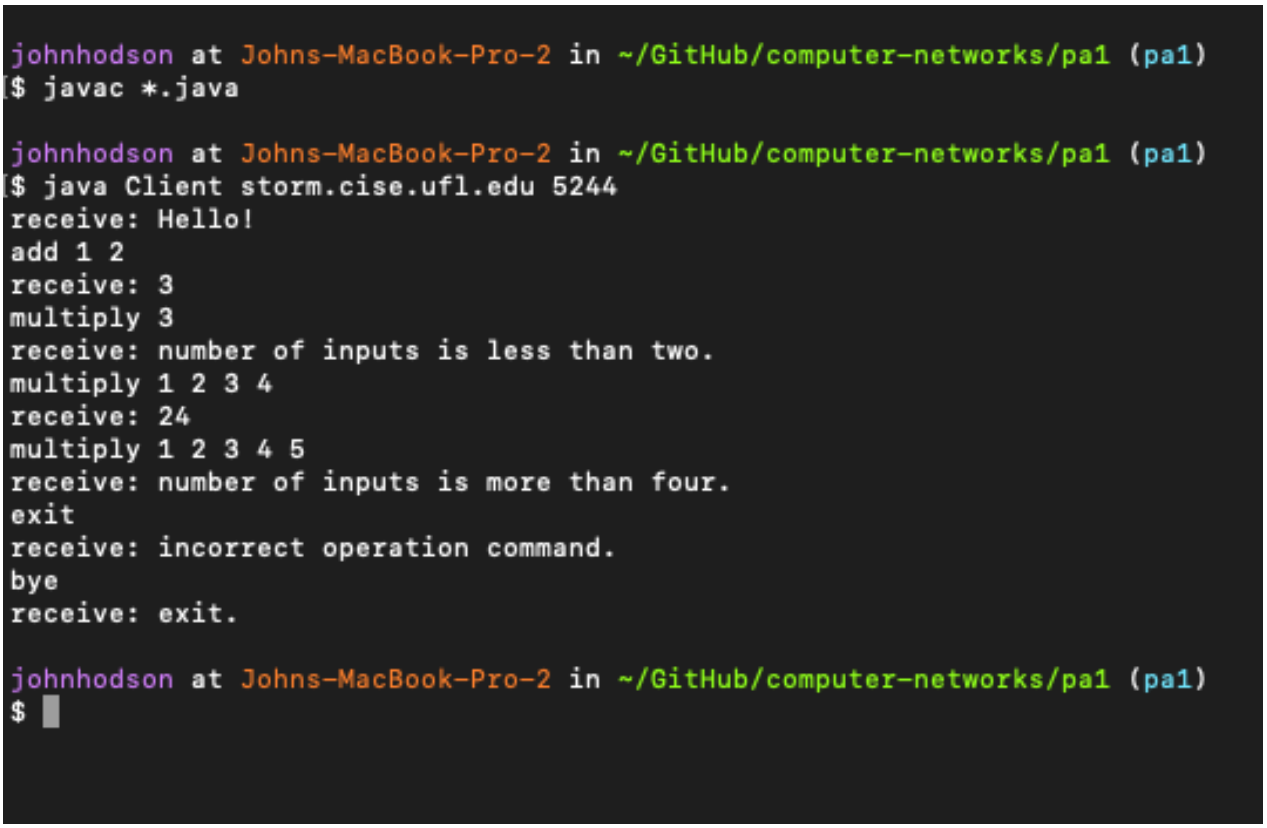
Both programs accept command line arguments. The server program accepts a port number on which to run. The client program accepts a host name and a port number to connect with. This is implemented as requested in the specification document.

The server program initiates a socket on this port and waits for a client connection. Once a client connects, it greets the client and then processes each of the client's requests until the client requests "bye," at which point the server sends "exit" and terminates the connection with the client. The client terminates at this point as well. If any client sends "terminate," then the server terminates as desired.

When the client program launches, it immediately attempts to connect to the server specified in the command line arguments. Note that this implies the server program must be running before the client program is launched.

I tested my program on the CISE thunder and storm servers. A discussion between my two programs can be seen in the two following pages, where I mimicked the conversation that was shown in the project document.

Client:


johnhodson at Johns-MacBook-Pro-2 in ~/GitHub/computer-networks/pa1 (pa1)
$ javac *.java
johnhodson at Johns-MacBook-Pro-2 in ~/GitHub/computer-networks/pa1 (pa1)
$ java Client storm.cise.ufl.edu 5244
receive: Hello!
add 1 2
receive: 3
multiply 3
receive: number of inputs is less than two.
multiply 1 2 3 4
receive: 24
multiply 1 2 3 4 5
receive: number of inputs is more than four.
exit
receive: incorrect operation command.
bye
receive: exit.
johnhodson at Johns-MacBook-Pro-2 in ~/GitHub/computer-networks/pa1 (pa1)
$

Server:

```
johnhodson at Johns-MBP-2 in ~/GitHub/computer-networks/pa1 (pa1)
$ sftp jhodson@thunder.cise.ufl.edu
jhodson@thunder.cise.ufl.edu's password:
Connected to jhodson@thunder.cise.ufl.edu.
sftp> cd CNT
Couldn't stat remote file: No such file or directory
sftp> cd cnt4007c
sftp> put Server.java
Uploading Server.java to /cise/homes/jhodson/cnt4007c/Server.java
Server.java
sftp> exit

johnhodson at Johns-MBP-2 in ~/GitHub/computer-networks/pa1 (pa1)
$ ssh jhodson@storm.cise.ufl.edu
jhodson@storm.cise.ufl.edu's password:
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-74-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

Authorized Access only
-------------------------------------------------------------
UNAUTHORIZED ACCESS TO THIS DEVICE IS PROHIBITED.
You must have explicit permission to access this
device. All activities performed on this device
are logged. Any violations of access policy can
result in disciplinary action.
-------------------------------------------------------------

Last login: Wed Feb  5 21:01:51 2020 from 24.136.57.187
storm:1% cd cnt4007c/
storm:2% javac *.java
storm:3% java Server 5244
get connection from /24.136.57.187
get: add 1 2, return: 3
get: multiply 3, return: -2
get: multiply 1 2 3 4, return: 24
get: multiply 1 2 3 4 5, return: -3
get: exit, return: -1
get: bye, return: -5
```

I was able to successfully implement every requested feature for programming assignment 1, with no bugs or limitations.

In the future, an abstract server class can be extracted from the server class that was implemented for this project in order to create a generic TCP server.