

# **Programming Assignment 1** **Report**

**Edward Tischler**  
**6842-1694**  
**[etischler@ufl.edu](mailto:etischler@ufl.edu)**

### **How to compile and run your code under which environment:**

I wrote and tested this code on sand.cise.ufl.edu (server) and storm.cise.ufl.edu (client). Rain (which is used in the example provided) I was unable to access. Therefore, run my code on the Linux systems that CISE provides. To compile, do `javac *.java`. To run my code, follow the style that was provided in the assignment description. For example, to run the server enter `'java server <port>'`. To run the client, enter `'java client <IP> <port>'`.

### **Description of code structure:**

Server:

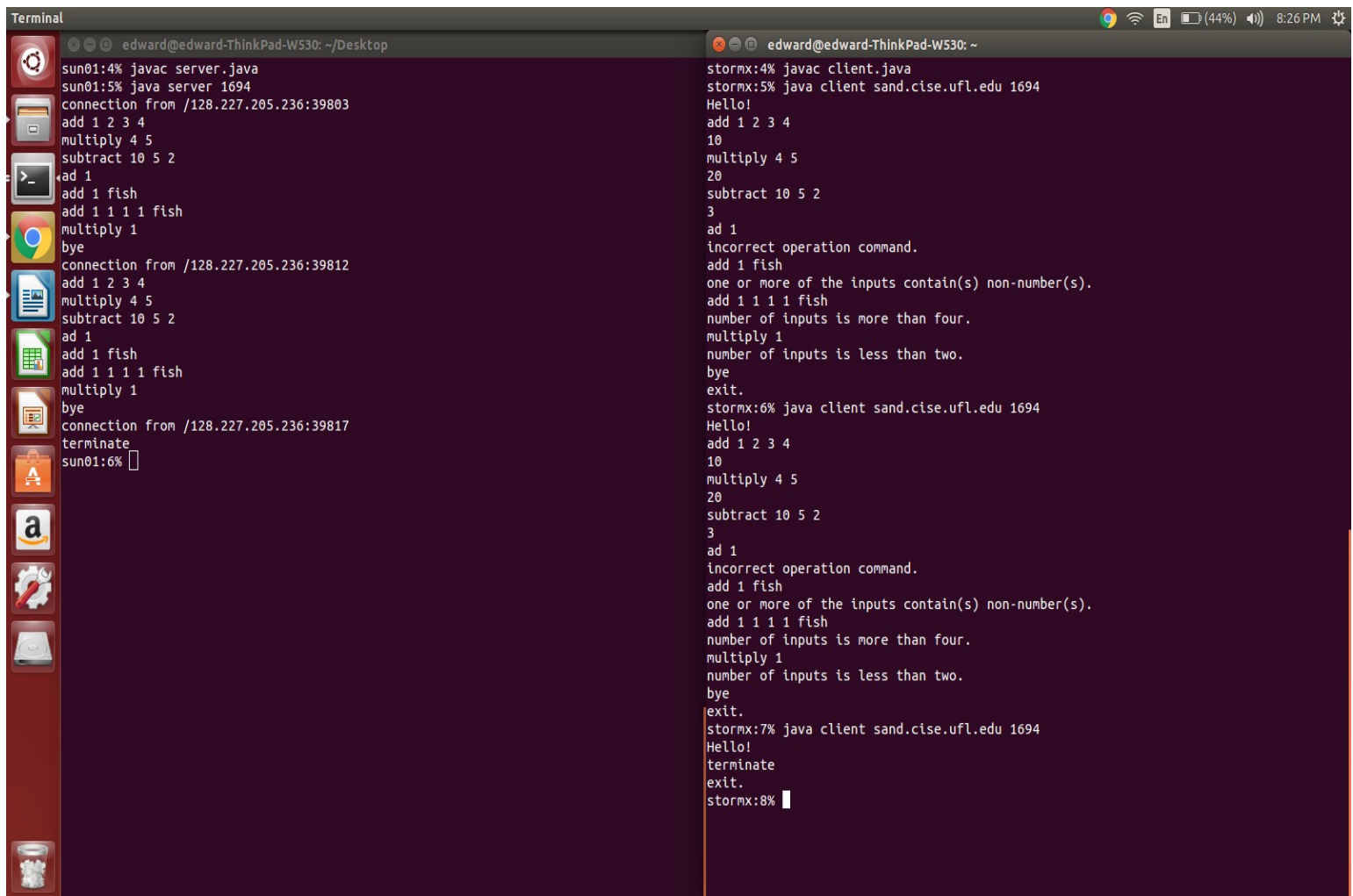
My server begins by opening a socket and awaits a new connection. Upon a new connection, my server send a 'Hello!' message which is to be read by the client. Afterwards in a while loop that awaits to see a terminate message, the server takes in input from the client. After it takes in input it separates words and numbers into arrays and removes punctuation. Afterwards it checks for the exceptions it has to handle. If an exception is found the program will reach the end of the while loop without executing anymore code and return the error code to the client. If it passes all the error checks which are placed in order of their precedence, it will then compute the result depending on the operation requested and send it back to the client. Upon a termination command being received, it will exit.

Client:

My client will also create a socket and attempt to connect to my server. Upon successful connection it will print the message received from the server. Then unless an -5 error code is received from the server, it consistently sends the client's message to the server and prints out the server's output on the client's screen. However, it will first check it against a bunch of error codes. If it passes the checks for error codes, it will print the result sent from the server based upon the requested computation.

//SCROLL DOWN FOR EXECUTION RESULTS

Show some of the execution results:



```
Terminal
edward@edward-ThinkPad-W530: ~/Desktop

sun01:4% javac server.java
sun01:5% java server 1694
connection from /128.227.205.236:39803
add 1 2 3 4
multiply 4 5
subtract 10 5 2
ad 1
add 1 fish
add 1 1 1 1 fish
multiply 1
bye
connection from /128.227.205.236:39812
add 1 2 3 4
multiply 4 5
subtract 10 5 2
ad 1
add 1 fish
add 1 1 1 1 fish
multiply 1
bye
connection from /128.227.205.236:39817
terminate
sun01:6%

stormx:4% javac client.java
stormx:5% java client sand.cise.ufl.edu 1694
Hello!
add 1 2 3 4
10
multiply 4 5
20
subtract 10 5 2
3
ad 1
incorrect operation command.
add 1 fish
one or more of the inputs contain(s) non-number(s).
add 1 1 1 1 fish
number of inputs is more than four.
multiply 1
number of inputs is less than two.
bye
exit.
stormx:6% java client sand.cise.ufl.edu 1694
Hello!
add 1 2 3 4
10
multiply 4 5
20
subtract 10 5 2
3
ad 1
incorrect operation command.
add 1 fish
one or more of the inputs contain(s) non-number(s).
add 1 1 1 1 fish
number of inputs is more than four.
multiply 1
number of inputs is less than two.
bye
exit.
stormx:7% java client sand.cise.ufl.edu 1694
Hello!
terminate
exit.
stormx:8%
```

Discuss the results you got with your program:

At first upon connection from the client to the server, the client receives a 'Hello!' response. I then show that the server awaits input from the client. I then show the correct arithmetic output coming to the client from the server based upon the client's input. I use different arithmetic operations and sizes of input to show that the code runs correctly. Next I run through all the possible error codes and show that they occur with precedence with the -1 error code having the most precedence. I then end the connection to the server and restart the connection. Afterwards, I rerun the same test as previously to show that multiple clients can connect to the server as long as it is one client at a time. After the second test is complete I show successful termination of both the client and the server when the client sends the terminate command to the server.

Discuss any abnormal results:

None

**Explain any bugs, missing items, and limitations of the program if there is any:**

I am not aware of any bugs, missing items, or limitations of my program.

**Any additional comments:**

None