You should save/rename this document using the naming convention HW2-MUid.docx (example: HW2-`johnsok9.docx`).

Objective: The objective of this exercise is to:
1. Work with sockets
2. Understand the socket communication process in Java and Linux tools

Submit: working java program and screen grabs showing tool and java operations

You **may** discuss the concepts with your fellow students
You **may not** show or share code with your fellow students
You **may not** show or share code with internet sources
You **may** discuss this with your instructor or TA.

Name:

Part #1: Socket Communications using Linux (ceclinux)

Estimated time: 1 hour

Exercise: Test the following protocol out using telnet, netcat and curl on Linux. Note, we mentioned telnet in class, but have NOT gone into the commands in depth. It is your job to review them and work out the usage. (hint: use the man command)

1. For each application (telnet, netcat, curl)
    a. Connect to 184.58.68.186 port 5001
    b. Send your muid followed by newline ('\n')
    c. Receive Greeting
    d. Send Formula (each line followed by newline)
        i. 10
        ii. +
        iii. 4
        iv. =
    e. Receive Answer (14.00)
    f. Exit
2. **Make a screen grab for each showing the operation including command line and the correct output**

Part #2: Program Modification and Testing

Estimated time: 1-3 hours

Exercise: This exercise will use your given socketClient.java program as the basis. Copy it and rename it hw2.java using the preceding (part 1) protocol to send math operations to a remote socket server and receive the correct response.

Find all TODO2 sections and replace/correct

1. Use command line arguments to run the program
    a. Do not use the hard coded address and muid from the lab
    b. java hw2 muid hostname port 10 + 5
2. Protocol
    a. Open socket to hostname/port (from command line)
    b. Send muid (from command line) (note all sends are terminated with newline)
    c. Receive greeting (display greeting as "greeting => " + greeting)
    d. Send Formula (from command line, by iterating over command line arguments)
    e. Send closing "="
    f. read response
    g. display formula created as "formula = >" + formula
    h. display response as "response => " + response
    i. exit
3. Example
    a. java SocketClient johnsok9 184.58.68.186 5001 10 + 5
    b. Open socket to 184.58.68.186 port 5001
    c. → "johnsok9\n"
    d. ← "Hello cse383 hw2: johnsok9\n"
    e. → "10\n"
    f. → "+\n"
    g. → "5\n"
    h. → "=\n"
    i. ← "15.00\n"
    j. Display (left justified, no tabs or extra spaces before words, however with the exact spacing between characters shown here. The Greeting is what is returned from the server. The Formula is what was sent along with the returned Answer.

```
Success
Greeting => Hello cse383 hw2: johnsok9
Formula  => 10+5
Answer   => 15.00
```

    k. Exit

SUBMIT: (15-60 minutes)

- Submit the working java program (hw2.java)
- Submit screen grabs (each one should show the command line used along with output)
    - Telnet
    - Netcat
    - Curl
    - Java
        - Show javac output that the program compiles
            - Should be a blank link
            - If javac prints anything – it is errors.
        - Show java runtime output that the program produces expected results