

You should save/rename this document using the naming convention **LabX-MUId.docx** (example: Lab2-johnsok9.docx).

Objective: The objective of this exercise is to:

1. Working with Git
2. Working with sockets
3. Basic I/O operations in java

Submit: working java program

You **may** discuss this with your fellow students or the instructor.

Name:

Part #1: Gitlab

Estimated time: 30 minutes

Exercise: This will use git to clone the given source code.

Complete this part of the exercise via the following steps:

Step1 Prepare Git

1. Sign into gitlab at https://gitlab.csi.miamioh.edu/users/sign_in using your miami uniqueid and password, verifying your account is ready. Stay logged in.
2. Log into the ceclinux machine using ssh
3. From the command line type
 - a. `ssh-keygen -t ed25519 -C "MUId@miamioh.edu"` (replace MUId with your uniqueid)
 - b. Will output the following (MUId will be your unique ID): (HIT ENTER UNTIL PROMPT SHOWS BACK UP)

```
Enter file in which to save the key
(/home/MUId/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in
/home/MUId/.ssh/id_ed25519.
Your public key has been saved in
/home/MUId/.ssh/id_ed25519.pub.
The key fingerprint is:
SHA256:Cg+INbn9NwtTynxEf/RyCOgVcXfcb0pF+OutHzUnDpk
MUId@miamioh.edu
The key's randomart image is:
+--[ED25519 256]--+
|                .+o. .++|
```

```

|   .   . .o o..+|
|   +   ... o .o.|
| o =   ... . *. +|
| . o +   S . E.o++|
|       B =   . o.o+|
|       X +   o..|
|       = o   .o|
|       .   .o.|
+-----[SHA256]-----+

```

4. Display the public created key with:
 - i. `cat ~/.ssh/id_ed25519.pub`
5. Copy the key by highlighting the output of cat with the mouse
6. Switch back to Gitlab on browser
7. Click on top right menu item - and select Profile
8. Now click on the pencil at the top right to edit the profile
9. Select SSH Keys on the left menu
10. Paste the copied public key into the key box on the right , then press “Add Key”

Step 2 Clone the source code (verifies if step1 is correct)

Create a classroom structure in your ceclnx01 system

1. `cd`
2. `mkdir cse383` (if it doesn't already exist)
3. `cd cse383`
4. `mkdir lab2`
5. `cd lab2`
6. `git clone git@gitlab.csi.miamioh.edu:johnsok9/cse-383-sockets.git` (note: use johnsok9 not your muid here)
7. Note: you should be working in your `~/cse383/lab2/cse-383-sockets` directory
8. Compile the socket program
 - a. `javac socketClient.java`
 - b. (should return with NO errors)

Part #2: Modify Code, Test and Submit

Estimated time: 30 minutes

Background

Exercise:

- The code you have just downloaded compiles but is not quite ready for the initial test.
- You need to update it using a text editor, then compile, run and verify
 - Edit (nano or vim)
 - `javac socketClient.java` to compile
 - `java socketClient` to run

1. Edit the file socketClient.java
 - a. Find every section with TODO1 and update it (including the top header)
 - Host/IP Address="184.58.68.186"
 - Port = 5002
 - MUID = your MUID
2. Compile the code
 - a. javac socketClient.java
3. Test the program
 - a. java socketClient
 - b. Should display:
Success
Greeting => Hello

SUBMIT:

- Submit the working java program
- Screen Grabs
 - Java compile showing program compiles with no errors
 - Java run – shows program runs (including command line executed) correctly
 - Git Web setup – Show screen grab from git assignment
 - Git clone worked
 - Screen grab showing output of “ls -l “ command with all files in the ~/cse383/lab2 directory