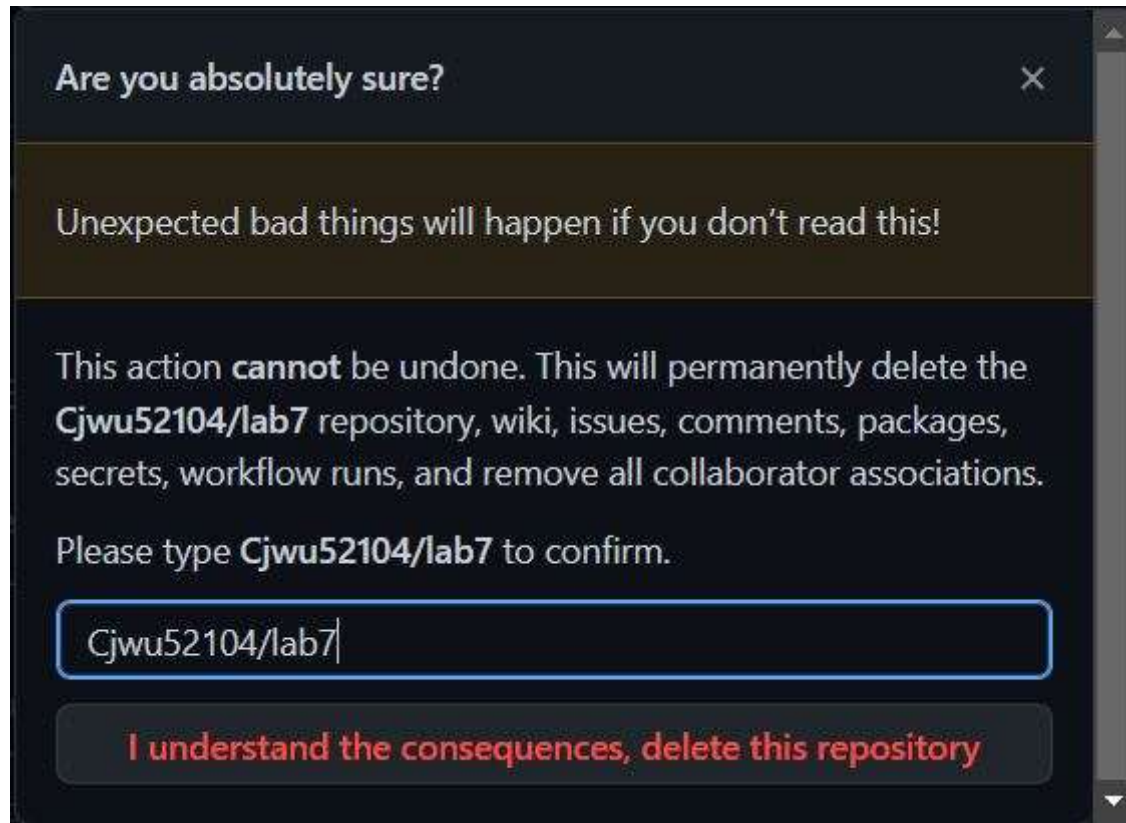# cse15l-lab-reports

Lab Report 4 - Connor Wu

1.) Delete fork

Cjwu52104/lab7

2.) Fork repo

https://github.com/ucsd-cse15l-w23/lab7

Fork > Create Fork

3.) Login with SSH

Keystrokes: `<ctrl>+r ssh <enter>`

4.) Clone

```
[cs15lwi23aox@ieng6-201]:~:141$ git clone git@github.com:Cjwu52104/lab7.git
Cloning into 'lab7'...
Warning: Permanently added the RSA host key for IP address '140.82.114.3' to the list of known hosts.
remote: Enumerating objects: 32, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 32 (delta 1), reused 2 (delta 1), pack-reused 27
Receiving objects: 100% (32/32), 371.75 KiB | 1.77 MiB/s, done.
Resolving deltas: 100% (11/11), done.
[cs15lwi23aox@ieng6-201]:~:142$ |
```

Keystrokes: `<ctrl>+r cl <enter>`

5.) Compile Tests + Run

```
[cs15lwi23aox@ieng6-201]:lab7:147$ javac -cp .:lib/hamcrest-core-1.3.jar:lib/junit-4.13.2.jar *.java
[cs15lwi23aox@ieng6-201]:lab7:148$ java -cp .:lib/hamcrest-core-1.3.jar:lib/junit-4.13.2.jar org.junit.runner.JUnitCore TestListExam
ples
JUnit version 4.13.2
..E
Time: 0.515
There was 1 failure:
1) testMerge2(TestListExamples)
org.junit.runners.model.TestTimedOutException: test timed out after 500 milliseconds
        at ListExamples.merge(ListExamples.java:43)
        at TestListExamples.testMerge2(TestListExamples.java:19)

FAILURES!!!
Tests run: 2,  Failures: 1
```

Keystrokes: `cd l<tab>, <ctrl>+r javac <enter>, <ctrl>+r ru <enter>`

6.) Fix Error

```
            index1 += 1;
        }
        else {
            result.add(list2.get(index2));
            index2 += 1;
        }
    }
    while(index1 < list1.size()) {
        result.add(list1.get(index1));
        index1 += 1;
    }
    while(index2 < list2.size()) {
        result.add(list2.get(index2));
        index2| += 1;
    }
    return result;
}


}
```

Keystrokes: `nano L<tab><enter>` then `<ctrl>+w return result <enter>`, `<ctrl>+w <enter><up><up>`
`<right(8x)><backspace>2, <ctrl>+o <enter> <ctrl>+x`

7.) Rerun tests

```
[cs15lwi23aox@ieng6-201]:lab7:144$ javac -cp .:lib/hamcrest-core-1.3.jar:lib/junit-4.13.2.jar *.java
[cs15lwi23aox@ieng6-201]:lab7:145$ java -cp .:lib/hamcrest-core-1.3.jar:lib/junit-4.13.2.jar org.junit.runner.JUnit
Core TestListExamples
JUnit version 4.13.2
..
Time: 0.017

OK (2 tests)

[cs15lwi23aox@ieng6-201]:lab7:146$ |
```

Keystrokes: `<ctrl>+r javac <enter> <ctrl>+r ru <enter>`

8.) Commit changes

```
[cs15lwi23aox@ieng6-201]:lab7:172$ git add .
[cs15lwi23aox@ieng6-201]:lab7:173$ git commit -m "fixed bug"
[main c549edc] fixed bug
 Committer: Connor J Wu <cs15lwi23aox@ieng6-201.ucsd.edu>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

 1 file changed, 1 insertion(+), 1 deletion(-)
[cs15lwi23aox@ieng6-201]:lab7:174$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 354 bytes | 177.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:Cjwu52104/lab7.git
   4a2b721..c549edc  main -> main
[cs15lwi23aox@ieng6-201]:lab7:175$ |
```

Keystrokes: `git add . <enter>, git commit -m "fixed bug", git push`

9.) Delete repo on ieng6

```
[cs15lwi23aox@ieng6-201]:lab7:160$ cd ..
[cs15lwi23aox@ieng6-201]:~:161$ rm -rf lab7/
[cs15lwi23aox@ieng6-201]:~:162$ ls
perl5   skill-demo1-data   skill-demo1-server   test   wavelet
[cs15lwi23aox@ieng6-201]:~:163$ |
```

Keystrokes: `cd .., rm -rf l<tab> <enter>, ls`

Shortcut References

`<ctrl>+r` is used to search previous commands, meaning that once you type a command out the first time, you can search for it with a few characters the second time and simply press enter to save time.

`<tab>` autofills the command/word you're typing if it's easily recognized by bash. For example, I did `nano L<tab> <enter>` in step 6 because there was only one file that started with L, so pressing tab completed the name of the file, ListExamples.java.

`<ctrl> + w` is a shortcut that allows you to search for string patterns while editing in nano. This allowed me to jump to the spot in the file containing the error, rather than pressing the down arrow numerous times to get there.

`<ctrl> + o` and `<ctrl> + x` save and exit a file (respectively) after editing it in nano. This makes it easy to edit files from the command lines, and to switch between editing and running commands, or perhaps another file.