

CSC207.01 2014S, Class 04: Unit Testing

Overview

- Preliminaries.
 - Admin.
 - Questions.
- A few thoughts on testing.
- An example: Testing exponentiation.
- A few notes on test-driven development.

Preliminaries

Admin

- I sometimes forget to update the lists of readings/labs and the link to the current reading.
 - You should let me know if you don't see them online.
 - You can always get them from the schedule.
- I *think* I've responded to all of the writeups I've received and publicly shamed those who didn't submit them.
- Mentor sessions Monday at 7:00 in 3821
- Lab writeup: Exercises 3f and 4c.
 - Remember that you are working with someone with a subsequent/prior last name (except in the case of SZ+EM+DNP)
 - Subject: CSC 207 Writeup 3: Unit Testing (YOUR NAME(s) HERE)
 - Due before class on Monday.
- Reading on classes for Monday's class.
- Notes from the review session are posted under eboards.
- HW 2 is due next Wednesday at 10:30 p.m.
 - Groups (2-4) required, but you can choose partners.
- Extra credit:
 - CS Table Friday at noon: 3D Printing (of body parts?).
 - Theatre Gigante Friday evening.
 - Anything else?
- Happy 40th anniversary Mac!

Questions

What's the normal order of operations in git?

```
git status - Figure out what you've modified.  
git diff FILE - Figure out how a file has changed.  
git add FILE - Indicate which files you want to commit  
git status - Figure out what you've modified.  
git commit - Commit those files, along with a message.  
git status (optional) - Make sure it succeeded  
git log (optional) - Review changes  
...  
git push - Send everything back to the server
```

How do I use vi?

vi is a modal editor. That means that you are either in insert mode (the mode in which you insert information) or editing mode (the mode in which you modify what's there or otherwise issue commands).

i puts you insert mode at the current location. Type while in insert mode.

The escape key gets you out of insert mode.

:wq saves and quits. (:w just saves, :q just quits)

Suppose I don't want to pop up an editor while using git commit. What should I do?

```
git commit -m "Message"
```

(Last semester, I said that I liked to open the editor each time. I've now gotten used to using -m.)

A few thoughts on testing

- Big message: Write programs to test, rather than testing by hand.
- The tools for testing are the easy part. Coming up with good tests is the hard part.

An example: Testing exponentiation

```
/**
 * Compute x^n
 * @pre
 *   n > 0
 *   x^n < Integer.MAX_VALUE
 *   x^n > Integer.MIN_VALUE
 */
int
expt(int x, int n)
```

- Talk to the people around you.
- Come up with a few tests.
- Idea: If it passes the tests, we're relatively certain it's correct. If it doesn't pass the tests, we know it's not correct.

Suggested tests

- Fractions
- Negative numbers
 - I expect that $(-2)^3 = -8$ `assertEquals("neg two cubed", -8, expt(-2, 3));`
- Positive numbers when exponentiated are divisible by themselves `assertEquals("divisible EM", 0, expt(3,7) % 3);`
- Special cases: Power of 0 `assertEquals("2^0 = 1", 1, expt(2,0));` for `(int i = -100; i < 100; i++) { assertEquals(i + "^0", 1, expt(i,0)); }` // for `(int i = -1000; i < 1000; i+= 10)` for `(int i = -100000; i < 1000000; i += Random.nextInt())`

Lab

Important lesson 1: You need `@Test` before any test you want to conduct.

Important lesson 2: Documentation is online at <http://junit.sourceforge.net/javadoc/org/junit/Assert.html>

Copyright (c) 2013-14 Samuel A. Rebelsky.



This work is licensed under a Creative Commons Attribution 3.0 Unported License. To view a copy of this license, visit <http://creativecommons.org/licenses/by/3.0/> or send a letter to Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, USA.