

15-122: Principles of Imperative Computation, Spring 2011

Assignment 0: Getting Started

William Lovas (wlovas@cs)

Out: Thursday, January 13, 2010

Due: Tuesday, January 18, 2010 (recommended)

Welcome to 15-122, Principles of Imperative Computation, Spring 2011 Edition! This ungraded “zeroth” homework is designed to give you a chance to get used to the course tools relevant to your success in 15-122 in a low-stress manner. Although it will not be graded, we encourage you to complete it so you can get used to some of the course workflow and work out any kinks before the first graded assignment.

1 Written: Bulletin Board (0 points)

Many course-relevant communiqués will be posted to the course *bulletin board*, or *bboard* (not to be confused with *Blackboard*, the web-based software we use for recording grades and administering quizzes). The bboard is called `academic.cs.15-122`, and you can subscribe to it via Andrew’s SquirrelMail Webmail client.

Exercise 1 (0 pts). Subscribe to the course bboard. Find the post entitled, “Homework 0: introductions” and respond to it with a short introduction of yourself, including your name, your major, and an answer to one of the following questions, your choice:

1. Who’s the computer scientist you most admire?
2. What’s your favorite movie featuring computers or computer science in a significant way?
3. What score do you predict for Saturday’s Steelers-Ravens match?

2 Programming: Exponentiation (0 points)

To get some experience writing and submitting C₀ code, we’ll work with the exponentiation example from the first two lectures. You can find this code on the [course website](#) under the “Schedule” link.

Exercise 2 (0 pts). Enter the corrected exponentiation code from Thursday's lecture as a function `int fastpow(int x, int y)` into a file `pow.c0`, including appropriate `//@`-annotations to document the function's pre- and post-conditions, loop invariants, and any assertions. (Remember to also include the `int pow(int x, int y)` function used in `fastpow`'s contract.) Submit this file using the command:

```
handin -a hw0 pow.c0
```

Be sure to test your code, but do not include a `main()` function in your submitted file. Instead, test your code by writing a `main()` function in another file, say `pow-testing.c0`, and compile both files together, e.g.,

```
cc0 pow.c0 pow-testing.c0
```

or

```
cc0 -d pow.c0 pow-testing.c0
```

(Remember that the `-d` flag enables dynamic checking of annotations.) Alternatively, you may test your code using the `coin` interpreter.¹

Exercise 3 (0 pts). Create a file `README.txt` explaining your code and its invariants briefly. Submit this file using the command:

```
handin -a hw0 README.txt
```

3 ... And That's It!

Once you've completed these exercises, you should be prepared to tackle homework 1, image manipulation!

¹We'll show you how in Friday's recitation.