CS 2150 Exam 1, fall 2015

Name   _____

You MUST write your e-mail ID on **EACH** page and bubble in your userid at the bottom of this first page. And put your name on the top of this page, too.

If you are still writing when "pens down" is called, your exam will be ripped up and not graded – even if you are still writing to fill in the bubble form. So please do that first. Sorry to have to be strict on this!

Other than bubbling in your userid at the bottom of this page, please do not write in the footer section of this page.

There are 6 pages to this exam. Once the exam starts, please make sure you have all the pages. Questions are worth different amounts of points.

If you do not bubble in this first page properly, you will not receive credit for the exam!

Answers for the short-answer questions should not exceed about 20 words; if your answer is too long (say, more than 30 words), you will get a zero for that question!

This exam is CLOSED text book, closed-notes, closed-calculator, closed-cell phone, closed-computer, closed-neighbor, etc. Questions are worth different amounts, so be sure to look over all the questions and plan your time accordingly. Please sign the honor pledge below.

_____

_____

_____

_____

The Tao that is seen
Is not the true Tao,
until You bring fresh toner.

(the bubble footer is automatically inserted into this space)

Page 2: C++

1. [3 points] In C++ programs, we always include the .h files (i.e., `ListNode.h`); why do you *not* include the cpp files (i.e., `ListNode.cpp`)?

2. [3 points] Explain *how* the `#ifndef FOO_H` / `#define FOO_H` / `#endif` pre-processor commands work.

3. [3 points] Consider a class `foo` that has a single (public) `int *p` field. There are no other methods, and only the default constructor created by the compiler. Write a C++ statement for each of the following steps: (1) dynamically allocate a new `foo` object, (2) set the `p` field in that object to a dynamically allocated int of value 42 (you can do this in two statements, if you'd like), and (3) print out that value (42) through the pointer declared in step 1.

4. [3 points] When are destructors called?

Page 3: C++, Lists

5. [3 points] Give example of a code segment that will *likely* cause a segmentation fault, but not definitely. Explain why it will or will not always cause a segmentation fault.

6. [3 points] Recall the `friend` keyword in C++. Why doesn't Java need that keyword?

7. [3 points] Which is a better way to implement a stack: with linked lists or with vectors? Briefly, why?

8. [3 points] Give three different examples of uses of queues.

Page 4: Numbers

9. [3 points] Convert 33 in base-6 to base-11. Show your work!

10. [3 points] Convert the binary 11011110 10101101 11000000 11011110 to hexadecimal.

11. [6 points] Encode -283 as a 16-bit, two's complement integer, and put the result in *little-Endian* hexadecimal.

Page 5: Numbers, Miscellaneous

12. [6 points] Convert -17.625 to it's 32-bit hexadecimal encoding using IEEE 754 floating point encoding. The result should be left in *little*-Endian format. Show your work!

13. [3 points] List 6 debugger commands. You can use either gdb or lldb commands, but all the commands you list must be from one or the other.

14. [3 points] List two things that Linux is better than Windows (or Mac OS X) for. List two things that Windows (or Mac OS X) is better than Linux for.

Page 6: No questions here

This page unintentionally left blank.

xkcd #135