CS 2150 Exam 1, spring 2016

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.

_____

_____

_____

_____

A crash reduces
Your expensive computer
To a simple stone.

(the bubble footer is automatically inserted into this space)

Page 2: C++

1. [3 points] Briefly describe what the `#include` command does in C++.

2. [3 points] What is reason we might use call by (or pass by) constant reference?

3. [3 points] Other than syntax, what are the three primary differences between references and pointers in C++?

4. [3 points] Write a snipped of C++ code (just a few statements) that creates a memory hole. Assuming the answer is correct, you will get more points for a shorter answer!

Page 3: Lists

5. [3 points] Is it more space efficient to implement a list with an array, or with a linked list? Briefly, why?

6. [3 points] Consider two array declarations: `int a[3];` and `int *b = new int[3];`. Other than the variable name, name two differences between `a` and `b`?

7. [3 points] Briefly explain why the `findKth(int index)` method is linear time for a linked list but constant time for an array.

8. [3 points] Consider the List class that you implemented in lab 2. Assume that the `count` field is removed, and calling `size()` requires re-computing the size each time. Write this new `size()` method.

Page 4: Numbers

9. [3 points] Convert 110010001 in binary (base 2) to octal (base 8). Show your work or explain how you did it.

10. [3 points] Imagine that we were designing a new floating point type, similar to the IEEE 754 format we studied in lecture; however, we have to pick the exponent and mantissa bit sizes. Briefly explain what advantage would we get from increasing the exponent bit size, and also what advantage we could get from increasing the mantissa bit size.

11. [6 points] Convert -4.6875 (which is $-4\frac{11}{16}$) to IEEE 754 floating-point notation, and express your answer as little-Endian. Show your work!
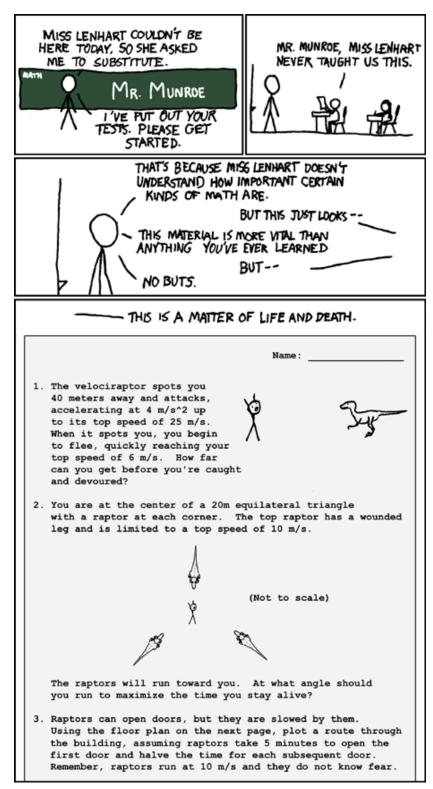
Page 5: Miscellaneous

12. [4 points] Suppose you need to run gdb on your program (a.out), set a breakpoint on line 31. At that point, you are in function `foo()`, which is called by `bar()`, which is called by `main()`. You want to view the values of two variables (x and y) that are in `bar()` (**NOT** `foo()`). Write the sequence of gdb (or lldb) commands that would allow you to do this entire operation.

13. [4 points] Suppose C++ stored arrays in column-major order instead of row-major order. Provide a formula that would compute the memory address location of `a[i][j]` under this scenario. Note that the declaration for `a` was: `double a[10][12];`.

14. [4 points] Consider a variation on a list called a "cyclic singly linked list", in which there is no head pointer, the tail pointer points to the tail node directly (no dummy), and the tail node points back to the first node. Write an `insertAtTail(int valueToInsert)` method for this type of linked list.

Page 6: No questions here

This page unintentionally left blank.

xkcd #135