CS 2150 Exam 1, spring 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] Why would we never bother to declare a reference as a variable *inside* a method (as opposed to as a parameter)?

2. [3 points] Why is a segmentation fault error called that?

3. [3 points] Assume that a pointer has been declared as: `int *i = new int(4);`. What happens to the *pointer* when you call `delete i;`? What are the restrictions on using the pointer after the `delete` line?

4. [3 points] Two common ways to cause segmentation faults are through `NULL` pointers (`int *i = NULL; cout << *i;`) and uninitialized pointers (`int *i; cout << *i;`). *How* do references prevent either of these from occurring?

Page 3: Lists

5. [6 points] Fill in the running times of the three primary operations in the various list-like data structures in the table below. To make it easier to grade, please just put an answer in the form such as "n", not "$\Theta(n)$" nor "linear".

Data structure	insert()	remove()	find()
List implemented via a vector			
List implemented via a linked list			
Stack implemented via a vector			
Stack implemented via a linked list			
Queue implemented via a vector			
Queue implemented via a linked list			

6. [3 points] Why are templates used in C++?

7. [3 points] List two real-world applications of stacks, and two real-world applications of queues.

Page 4: Numbers

8. [6 points] Convert -2.21875 ($-2\frac{7}{32}$) to IEEE floating-point notation. Your answer should be shown in big-Endian hexadecimal. Be sure to show all your work!

9. [4 points] Convert -373 to 32-bit two's complement hexadecimal notation. Your answer should be shown in *little*-Endian.

10. [2 points] Why is little-Endian used? Meaning, why to some computers represent their numbers using little-Endian?

Page 5: Miscellaneous

11. [3 points] Imagine that you just compiled a C++ program, called `a.out`. You want to run that program, and use the contents of the file named `input.txt` as the input (i.e., what `cin` will read), and save the output in a file named `output.txt` (i.e., what `cout` will print). Write a *single* UNIX command line that will accomplish this.

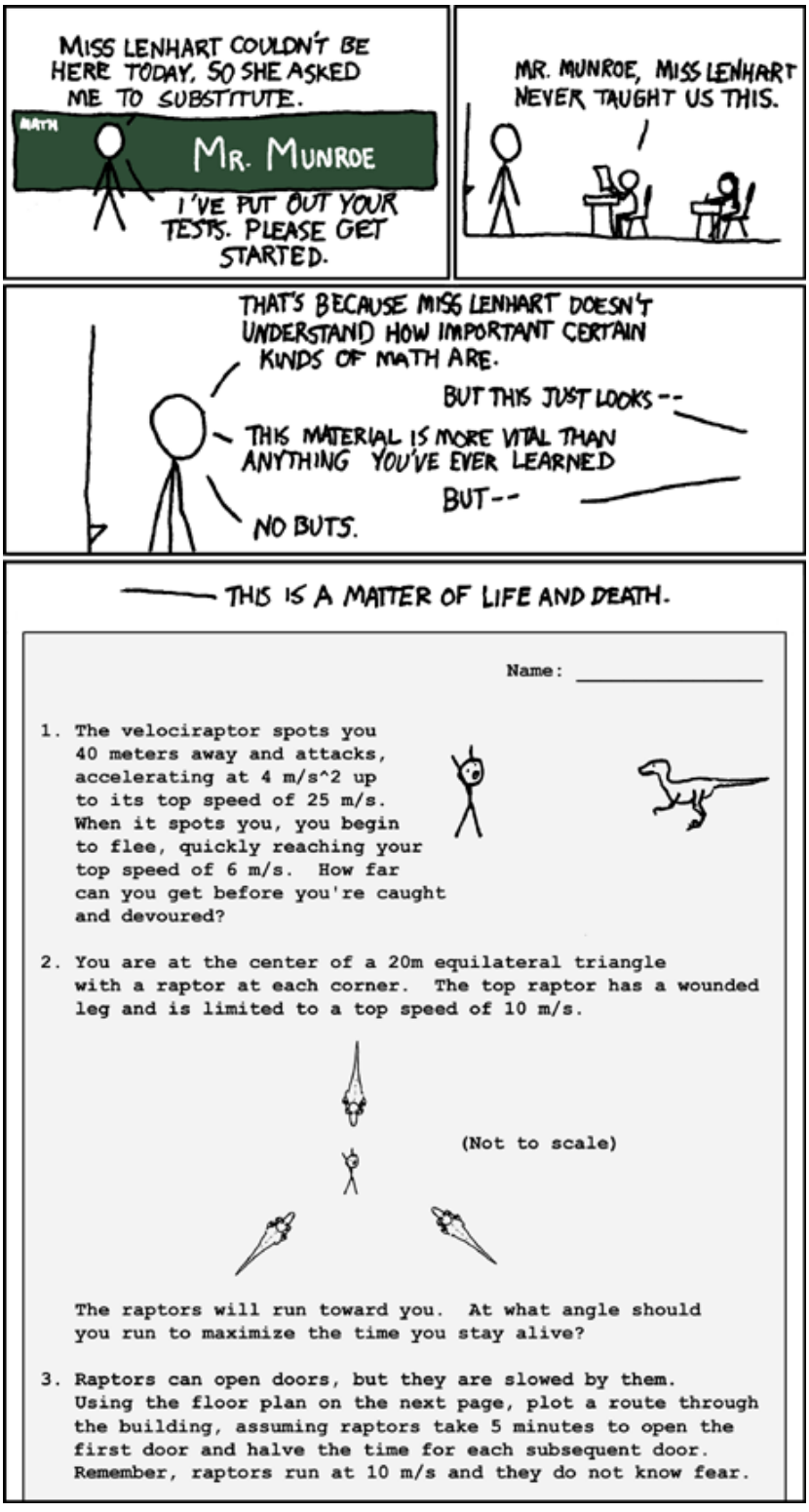
12. [3 points] Why do we make you learn and use Unix / Linux in this course?

13. [3 points] Why do we like big-Theta instead of big-Oh?

14. [3 points] How much do you wish we had a snow day today?

Page 6: No questions here

This page unintentionally left blank.



xkcd #135