

CS 2150 exam 2, fall 2014

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.

*Serious error.
All shortcuts have disappeared.
Screen. Mind. Both are blank.*

(the bubble footer is automatically inserted into this space)

Page 2: Old Stuff

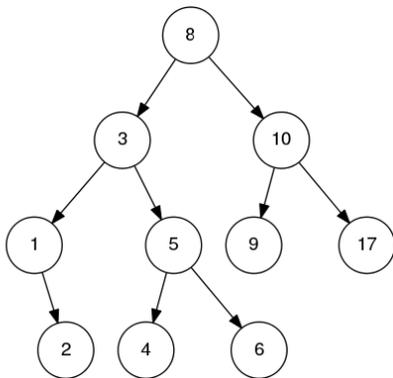
1. [3 points] Why does the C++ compiler have such a hard time determining errors with template code?
2. [3 points] Write a C++ function that returns the 13th bit of a passed `int` parameter.
3. [3 points] What does the `backtrace` command do in `gdb`?
4. [3 points] Assume that `Rational *r = new Rational()` has been declared. Explain *exactly* what happens when `delete r` is called.

Page 3: Trees

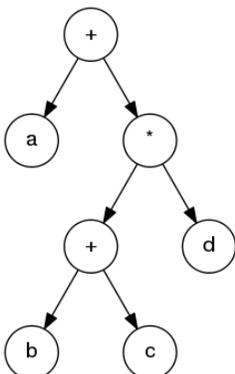
5. [3 points] Give three reasons when one would you *not* want to use trees as a data structure.

6. [3 points] Describe the *splay()* algorithm used in splay trees.

7. [3 points] Insert 7 into the AVL tree shown below. Show the resulting tree.



8. [3 points] Write the pre-order, in-order, and post-order traversal of the following tree. Do not include parenthesis in any of the traversals.



Page 4: Hashes

9. [3 points] What is the best way to handle deletes in a hash table? Why?
10. [3 points] Why are linked lists always used for the separate chaining buckets over a more efficient data structure, such as a balanced binary tree?
11. [6 points] In the tables below, you are to perform the specified inserts on the hash tables. You should perform the operations listed above the table, and show the results in the table below. Note, however, that each of the tables assume that you start with the table immediately to the left, and then perform the operations listed above that table (so, for example, the 2nd table, under the "insert 30", should have 20, 24, 28, *and* 30 inserted into it). The hash table uses double hashing as its collision resolution strategy. The primary hash function is $h(k) = k \bmod 10$, and the secondary hash function is $h_2(k) = (k \bmod 6) + 1$.

insert 20, 24, 28:

0	
1	
2	
3	
4	
5	
6	
7	
8	
9	

insert 30:

0	
1	
2	
3	
4	
5	
6	
7	
8	
9	

insert 44:

0	
1	
2	
3	
4	
5	
6	
7	
8	
9	

insert 38:

0	
1	
2	
3	
4	
5	
6	
7	
8	
9	

