CS 216 Final Exam

You MUST write your name and e-mail ID on EACH page and bubble in your userid at the bottom of EACH page – including this page.

If you do not do this, you will receive a zero for that page! (Or a grade penalty if you leave out the bubble form for this page)

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 forms. So please do that first. Sorry to have to be strict on this…

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

There are 10 pages to this exam – once the exam starts, please make sure you have all 10 pages.

There are three types of questions: short answer (worth 5 points each), medium answer (worth 10 points each), and long answer (worth 20 points each). The short answer questions should not take more than a line or two to answer – *your answer should not exceed about 20 words.* There are 180 points of questions and 180 minutes (3 hours) to take the exam, which means you should spend about a minute per question point.

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 here:

____

____

____

____

There are 10 types of people in the world – those that understand binary and those that don't.

First exam stuff

1. [10 points] Other than syntax (i.e., the C++ code snippets used to declare and use them), what are all the ways that references differ from pointers in C++?

2. [10 points] IBM decided, in an effort to be incompatible with everybody (including themselves), to not use the IEEE 754 floating point standard, but instead to define their own. This standard is the same as IEEE 754, except that it uses 7 bits for the exponent and 24 for the mantissa. Note that, like IEEE 754 floating point, the offset for the exponent is still 2^{n-1}-1, where *n* is the number of bits of the exponent. Using IBM's standard, convert 10.5 to hexadecimal notation, leaving the result in big-endian format. For simplicity, we'll assume that everything after the 5th bit in the mantissa is 0.

Second exam stuff

3. [5 points] What does NP-complete mean? Why do we care?

4. [5 points] For each of the three balanced tree types that we studied (AVL, red-black, and splay), list one reason that we would use that particular tree type over the other tree types.

5. [10 points] So people in this class so loved IBCM, that they decided to write a compiler module that would compile C++ code into IBCM code. But there are one or two (or a thousand, really) challenges that we face in doing so. In particular, we need a stack to hold activation records for subroutines. How might we go about implementing a stack within IBCM?

----

x86, page 1

6. [20 points] Intel's new 32-bit chip just came out! It's called the Hextium, and it's *such* an improvement over the Pentium. In particular, it has more general purpose registers – in addition to the general purpose registers of the Pentium (edi, esi, eax, ebx, ecx, and edx), it has 4 more (eex, efx, egx, ehx). The Hextium still has esp and ebp, of course. It has been decided that the four additional registers will be used to pass parameters to a subroutine – if there are more than 4 parameters, any additional ones are placed on the stack. The rest of the calling convention is similar to the one we studied. Describe the full calling conventions (prologue and epilogue for both the caller and the callee) for the Hextium with these new registers. Keep in mind that your convention must work for recursive functions!

Heaps

7. [5 points] What is the binary heap ordering property? What is the binary heap structure property?

8. [5 points] What are the big-Oh running times of the main heap operations (insert, deleteMin)? What are their big-theta running times?

9. [10 points] In a delete min operation in a heap, we replace the root with the "last" value in the heap as per the structure property. We then percolate that value downwards as necessary, replacing that value with the smaller of its two children. Why with the smaller one? Give an example of when replacing it with the larger of the two children would cause a problem.

Huffman Coding

10. [10 points] Given the following characters and their respective occurrence counts, what is the Huffman coding tree that is built from this data? Use scrap paper to build up the tree – we are only interested in the final tree here.

s	6
n	7
o	8
r	6
k	1
e	13
l	4

Total letters: 45

11. [10 points] What is the algorithm for decoding of a Huffman coded message? We are looking for an outline of the steps, not C++ code.

Graphs

12. [5 points] Describe briefly, but concisely, the algorithm for finding the topological sort of a graph.

13. [5 points] Describe briefly, but concisely, Prim's algorithm for finding the minimal spanning tree.

14. [5 points] Describe briefly, but concisely, Kruskal's algorithm for finding the minimal spanning tree.

15. [5 points] The best known shortest path algorithm takes O(*e* log *v*) time, once optimized. Obviously, Google maps does not take this much time to determine a route, as that would still take too long with the huge data set that they use (all the roads in the US). What optimizations might they use?

Miscellaneous

16. [5 points] In aspect-oriented programming, what is a join point? What is a point cut?

17. [5 points] Name a unique feature of Intercal, meaning one that is *not* in any other known programming language that you know.

18. [5 points] What is the big-Oh running time of the algorithm that we used to implement the traveling salesman problem? How would you decrease the big-Oh running time?

19. [5 points] What is lossy compression? Give an example.

(the bubble footer is automatically inserted in this space)

Memory

20. [5 points] Describe briefly, but concisely, how a buffer overflow attack works in C/C++.

21. [5 points] Why are cache's important? Why do we need them?

22. [5 points] What is temporal locality? What is spatial locality?

23. [5 points] Write a VERY SHORT code snipet in C/C++ that does not take advantage of caches (meaning one that works against the cache, not with it).

(the bubble footer is automatically inserted in this space)

UNIX

24. [5 points] What do pipes do in Unix?

25. [5 points] Why would we use shell scripting? Meaning, when is it more useful than any other programming language?

26. [10 points] Write a makefile that will compile a program that has three C++ source files: foo.cpp, bar.cpp, and qux.cpp. The final executable should be called thud (or thud.exe, your choice). Also include a clean target that removes the object files.

(the bubble footer is automatically inserted in this space)