

CS 2150 Exam 1, fall 2021

Name _____

You **MUST** *clearly* write your name above. You must also write your e-mail ID on **EACH** page.

If you are still writing when “pens down” is called, your exam will not be graded – even if you are still writing to fill your name and userid. So please do that first. Sorry to have to be strict on this!

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.

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.

*You step in the stream,
But the water has moved on.
This page is not here.*

Page 2: C++

1. [6 points] Consider the following three compile-time error messages. For *each* of these three error messages, *briefly* describe what is wrong, and how you would fix it.

```
xam1_error.cpp:8:11: error: use of undeclared identifier 'b'  
  cout << b << endl;  
         ^
```

```
exam1_error.cpp:11:3: error: no matching function for call to 'do_function'  
  do_function(s);  
  ^~~~~~
```

```
exam1_error.cpp:4:6: note: candidate function not viable: no known conversion  
from 'std::string' (aka 'basic_string<char>') to 'int' for 1st argument  
void do_function(int x) {  
  ^
```

```
exam1_error.cpp:16:5: error: redefinition of 'main'  
int main() {  
  ^
```

```
exam1_error.cpp:7:5: note: previous definition is here  
int main() {  
  ^
```

3 errors generated.

2. [3 points] *Briefly*, why can't you make C++ segfault with references? *HINT: Think about the three rules of references and why they help prevent segfault errors.*

3. [3 points] *Briefly*, what is the main difference between a pointer and a (statically-allocated) array base name? *Briefly*, why does that difference exist?

Page 3: Lists

4. [3 points] Consider a *singly* linked list with just a head pointer in the `List` class. This linked list does not have any dummy nodes. Each `ListNode` has two fields: `int value` and `ListNode *next`. *Briefly*, what does the following code do if given the pointer to the head node?

```
void mystery(ListNode *l) {
    if ( l ) {
        mystery(l->next);
        cout << l->value << endl;
    }
}
```

5. [3 points] *Briefly*, give two reasons when you would want to use an array based (not vector based!) list instead of a linked-list based list.

6. [3 points] *Briefly*, give two reasons when you would want to use a linked-list based list instead of an array based (not vector based!) list. Note that you used a given reason in the previous question, you can't use the opposite reason here!

7. [3 points] What is the **worst-case** big-Theta running time of `vector::push_back()`? *Briefly*, why?

Page 4: Numbers

8. [6 points] Consider the following code. What is the (big-Endian) hexadecimal value stored in variable `c` at the end of the `main()` function? Show your work!

```
int main() {  
    float a = -6.00;  
    float b = -4.75;  
    float c = a + b;  
    return 0;  
}
```

9. [6 points] Consider the statement `int x = -66085;`. Note that $66085 = 65536 + 512 + 32 + 4 + 1 = 2^{16} + 2^9 + 2^5 + 2^2 + 2^0$. Also note that the value stored in the variable is negative. Assume that `int x` is a 32-bit two's complement integer. What is the *little-Endian* representation of this value in hexadecimal? Show your work!

Page 5: Miscellaneous

10. [3 points] When we compile our programs, we often use two flags: `-Wall` and `-g`. *Briefly* explain what each of these flags do.

11. [4 points] Consider an array declared as `int a[4][4]`; with ints that are 4 bytes in size. You know that the address of `a[2][3]` is `0xff998308`. What, then, is the address of `a`? Note that all these addresses are in big-Endian.

12. [3 points] If you were designing your own floating-point type, imagine that you had a fixed number of bits to fit it into (32, 64, etc.). *Briefly* describe why you would want more mantissa bits versus why you would want more exponent bits.

13. [2 points] Have you completed the UNIX honor pledge? If not, when will you complete it by? FYI, the link to the UNIX honor pledge is on the Collab landing page. (You get full credit on this question as long as you answer honestly.)

Page 6: No questions here

This page unintentionally left blank.

