CS 2150 Exam 1

Name _____

You MUST write your e-mail ID on **EACH** 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 – 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. [4 points] What is the output of the following C++ code? This code does in fact run without a seg-fault. Hint: Remember that arrays in C++ are stored in row-major order!

```
// Assume #includes and using namespace appear up here somewhere
int main()
  int arr[3][4] = {0} // fill the array with 0's
  arr[0][7] = 7;
  arr[2][2] = 4;
  arr[1][5] = 6;
  cout << arr[1][3] << arr[0][10] << arr[1][7] << arr[1][5] << endl;
  return 0;
}
```

2. [3 points] Briefly describe the difference between the copy construction and operator=() functions?

3. [3 points] Give a small (1-3 lines) code snippet that **might** produce a seg-fault in C++. Briefly describe why this is the case.

4. [3 points] Suppose you want to write a method that takes two linked lists, a and b, that concatenates the two lists. Afterwards, both a and b would contain the contents of both lists (first the elements of a followed by those of b). Briefly describe your approach for implementing this method. What is the runtime of your algorithm and why?

Page 3: Queues!

Suppose we are implementing a linked-list based **Queue** class that only supports `int`s:

```
class Queue {
public:
    Queue();
    ~Queue();
    Queue(const Queue& other);
    void enqueue(int arg);
    int dequeue();
    bool isEmpty() const;
private:
    int *entries;      //dynamically allocated array of ints
    int size;          //number of elements in this array
    int frontIndex;    //index of the front element in the queue
    int backIndex;     //index of the back element in the queue
};
```

Note that a queue with `backIndex` equal to `frontIndex` - 1 is always considered empty.

5. [4 points] Which of the following code snippets should appear in the implementation of Queue::~Queue? **Circle all that apply.**

```
delete entries;

delete[] entries;

entries = new int[size];

entries = new int[backIndex + 1];
```

6. [4 points] Which of the following code snippets would likely be part of some correct implementation of Queue::Queue(const Queue& other)? **Circle all that apply.**

```
size = other.size;

entries = new int;

entries = new int[other.size];

entries = other.entries;

delete[] other.entries;

frontIndex += 1;

frontIndex = backIndex;
```

7. [3 points] If one has Queue a; and Queue b; and try to do b = a; one will get an error with the above declaration of the `Queue` class. Adding a method to the above class would eliminate this error. What is the signature (name, argument types, return types) of that method?

Page 4: Stacks

For this question, consider a stack implemented with a singly-linked list:

```
class StackNode {
private:
    int value;
    StackNode *next;
    friend class Stack;
};
```

```
class Stack {
public:
    Stack();
    Stack(const Stack &other);
    // ... some methods omitted ...
    int top();
    void push(int value);
    void pop();
private:
    StackNode *top;
};
```

8. [4 points] Consider the following implementation of pop()

```
void Stack::pop() {
    top = top->next;
}
```

Which of the following are true. **Circle all that apply.**

- It should be a const method (and so defined as "**void** Stack::pop() **const**").
- It leaks memory.
- It should use top.next instead of top->next.
- It should make top point to a new StackNode instead of an existing one.
- It should run **delete** top->value;

9. [3 points] Does this Stack class need a destructor other than the default one supplied by the compiler? Explain briefly.

10. [3 points] Consider the following **broken** implementation of Stack::Stack(**const** Stack&) and a corresponding invocation:

```
Stack::Stack(const Stack& other) {
    top = other.top;
}

//somewhere else in the program, the method above is invoked like this:
Stack secondStack(firstStack);
```

Let's suppose one discovers that `secondStack` behaves correctly, but if one uses `secondStack` after the program has done other tasks, it contains the wrong values and sometimes even segfaults. Explain how the implementation of Stack::Stack(**const** Stack&) would be responsible for this problem.

Page 5: Can I have your number?

11. [4 points] Please select True or False for the following statements:

    T / F 678_9 is equal to 0x22d and also 557_{10}

    T / F 729 is a valid number in base 6.

    T / F The number of distinct IEEE 754 32-bit floats is greater than those for a 32-bit unsigned int

    T / F Regardless of convention, 16-bit floats always have smaller max values than 32-bit floats

    T / F 101_b is equal to 0x5

12. [3 points] Consider a IEEE-like floating point format with:

    - 1 sign bit;
    - 3 exponent bits; and
    - 4 mantissa bits

    Assume we follow a similar convention for exponent bias, special / reserved values, and mantissa representation. For this question, write those bits in this order, most significant bit first.

    What is the bit representation of 2.125 in this floating point format?

13. [3 points] What is the largest non-infinite possible value in this floating point format? Write your answer as an equation as a base-10 number or an equation involving only base-10 numbers.

14. [3 points] For these questions, consider 10-bit two's complement numbers.

    What is the most positive such number? Write your answer as a base 10 number.

15. [3 points] What is the most negative such number? Write your answer as a base 10 number.

16. [3 points] What number is represented by the bits `11111 11100`?

Page 6: Big-OH!

17. [3 points] In the formal definition of Big-Oh, what is the purpose of the constant c? Why do we need it? Give an example to illustrate your point.

18. [3 points] In the formal definition of Big-Oh, what is the purpose of the constant n_0? Why do we need it? Give an example to illustrate your point.

19. [3 points] Suppose I want to write a method in my Vector class that inserts an element at exactly the 10th position from the end. Your algorithm will shift the last 9 elements each and insert the new element at the 10th position. What is the big-theta runtime of this algorithm? Why? You may ignore the case where the Vector needs to grow in size. We are looking for the runtime on a common insert in which the Vector doesn't need to grow.

20. [3 points] Suppose I want to write a method in my Vector class that inserts an element at exactly the $\frac{n}{10}$ position from the end. Like before, your algorithm will shift elements out of the way and then insert into the gap in the Vector. What is the big-theta runtime of this algorithm? Why? You may ignore the case where the Vector needs to grow in size. We are looking for the runtime on a common insert in which the Vector doesn't need to grow.