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 the line where the sky meets the C, it caaaaalls me

1. [3 points] What is the difference between the following two function declarations? Which is generally preferred and why?

```
void f(string s)
void f(const string& s)
```

2. [3 points] List two differences between static and dynamic memory allocation.

3. [3 points] Consider the following code. What is printed?

```
int n = 2;
int *p = &n;
*p = 5;
n += 1
cout << n << end;
```

4. [3 points] What is the purpose of the destructor? List the two ways in which the destructor is invoked.

Page 3: Linked Lists!

The following questions are about Lab 2: Linked Lists. Assume that all appropriate files/libraries are #included.

5. [5 points] The following statements pertain to the code snippet on the right. **Circle all that apply**.

a) This code segment will not compile.
b) This code segment will not properly initialize the head and tail.
c) This code segment will not properly initialize count.
d) The head and tail ListNodes should not be dynamically allocated.
e) "List::" should not precede the method name.

```
List::List() {
    ListNode *head = new ListNode();
    ListNode *tail = new ListNode();
    head->next = tail;
    tail->previous = head;
    count = 0;
}
```

6. [5 points] The following statements pertain to the code snippet on the right. **Circle all that apply**.

a) This code segment will not compile.
b) This code segment will always segfault.
c) This code segment will exhibit undefined behavior, but is likely to work correctly because of how memory deallocation works.
d) "pos.current" should be changed to pos->current, and pos.moveForward() should be changed to pos->moveForward().
e) The line "delete pos;" needs to be added once we're done using pos to avoid memory leaks.

```
void List::makeEmpty() {
  ListItr pos(head);
  pos.moveForward();
  while (!pos.isPastEnd()) {
    pos.current->next->previous = head;
    delete head->next;
    pos.moveForward();
    head->next = pos.current;
  }
}
```

7. [5 points] The statements in this method are out of order! Re-order lines 1-4 to fix the method by writing the labels 1-4 in the correct order. (There may be more than one correct order. In that case, only write one solution). Please write your answer as a list of numbers (e.g., 1-2-3-4).

```
  void List::insertAfter(int x, ListItr position) {
    ListNode *insert = new ListNode();
    insert->value = x;
(1) insert->previous = position.current;
(2) position.current->next = insert;
(3) insert->next->previous = insert;
(4) insert->next = position.current->next;

    count++;
  }
```

Page 4: Miscellaneous

8. [3 points] Suppose you would like to insert a value into a vector. List two situations in which this insertion will take linear time, and one situation in which it will take constant time.

9. [3 points] When formally defining $O(g(n))$, what is the purpose of n_0?

10. [6 points] The following code shows the enqueue() and dequeue() operations implemented using a double-linked list. The doubly-linked list used here DOES NOT have dummy head and tail nodes. Fill in the missing code to complete the methods.

```
void Queue::enqueue(int x) {

  ________________ = new QueueNode(x);

  newNode->next = this->head;

  newNode->previous = ____________;

  ____________________ = newNode;

  this->head = newNode;
}
```

```
int Queue::dequeue() {

  QueueNode* toRemove = this->tail;

  int toReturn = toRemove->value;

  ____________ = toRemove->previous;

  this->tail->next = NULL;

  ________________________;

  ________________________;
}
```

Page 5: Can I have your number?

11. [3 points] Suppose we decided to represent floating point numbers using a *fixed-point notation* in which we use the first 9 bits as a *two's complement whole number* and the next 23 bits as *fractional bits* (i.e., each position i from 1-23 represents an additional $\frac{1}{2^i}$). What is the highest value we can represent? Is this higher or lower than the IEEE 754 Floating Point convention we saw in class?

12. [3 points] Convert -19 to a 12-bit two's complement number. Leave your answer in *big-endian* hexadecimal.

13. [6 points] Convert -117.25 to an *IEEE 754 Floating Point Number*. Leave your answer in *little-endian* hexadecimal.

Page 6: This page intentionally left blank

You may use this space for scratch work, however **nothing you write on this page will graded**.