**CMSC 206: Data Structures**

**Iterators, Stacks, Queues, and Priority Queues**

* 1. For each labeled line of code below, draw the contents of the linked list list after that line is run. Underline the element whose data is the most recently returned from the next method of itor, if applicable.

LinkedList<String> list = new LinkedList<String>();

list.addLast("a");

list.addLast("b");

list.addLast("c");

1. list.addLast("d");

ListIterator<String> itor = list.listIterator();

1. itor.next();
2. itor.remove();

itor = list.listIterator();

1. itor.next();
2. itor.next();
3. itor.set("e");
4. itor.add("f");
5. itor.next();
6. itor.add("g");

1. After each labeled line, write out what the contents of the stack are. List the contents in order from next-to-be-popped to last-to-be-popped:

Stack<Integer> s = new Stack<Integer>();

s.push(3);

* 1. s.push(4);
  2. s.pop();

s.push(10);

s.push(12);

* 1. s.push(8);
  2. s.push(s.pop() + s.pop());

1. After each labeled line, write out what the contents of the queue are. List the contents in order from next-to-be-dequeued to last-to-be-dequeued.

Queue<Integer> q = new LinkedList<Integer>();

q.add(3);

1. q.add(4);
2. q.remove();

q.add(10);

q.add(12);

1. q.add(8);
2. q.add(q.remove() + q.remove());