The Midterm

- Closed book, closed notes, etc., but the first page will list useful APIs (List/ArrayList/ArrayIntList/LinkedListIntList)
- 25% of your grade = 5 homework assignments
- Covers everything we’ve discussed in lecture and homework
- Study strategy:
  - Midterm and final from last quarter
  - Practice-It problems
  - Study your own notes and the online slides
  - Textbook
- My midterms tend to be challenging and long.
Exam Question Types

1. Reading: Polymorphism puzzle (EQ1)
2. Reading: List/ArrayList usage (EQ2)
3. Reading: Reference manipulation (EQ3)
4. Reading: GUIs (layouts and listeners)
5. Writing: Implement inheritance and Comparable interface
6. Writing: Linked list operations (EQ4)
7. Writing: Using lists (EQ5)

No questions on exceptions or iterators.
Example Question 1

public class Do {
    public void m1() { System.out.println("A1"); }
    public void m2() {
        m1(); System.out.println("A2");
    }
}

public class Re extends Do {
    public void m1() { System.out.println("B1"); }
    public void m3() { System.out.println("B3"); }
}

public class Mi extends Re {
    public void m1() {
        m3(); System.out.println("C1");
    }
    public void m2() {
        super.m2(); System.out.println("C2");
    }
}

Suppose the variables are defined:

Do do1 = new Do();
Do do2 = new Re();
Do do3 = new Mi();
Object obj1 = new Do();
Object obj2 = new Re();

What do the following statements do? (Write “error” for errors.)

do1.m2();
do2.m2();
do3.m1();
do3.m2();
do3.m3();
((Re) do3).m3();
((Re) do3).m1();
obj1.m1();
((Re) obj1).m2();
((Do) obj2).m1();
public static void mystery(List<Integer> list) {
    for (int i = 0; i < list.size(); i++) {
        int n = list.get(i);
        if (n % 10 == 0) {
            list.remove(i);
            list.add(n);
        }
    }
    System.out.println(list);
}

Write the output for the following lists inputs:
[1, 20, 3, 40]
[80, 3, 40, 20, 7]
[40, 20, 60, 1, 80, 30]
```
ListNode x = new ListNode(1);
ListNode y = new ListNode(2, x);
ListNode z = new ListNode(3, y);
x.next = z;
y.next.next = y;
z = x;
x.data = 4;
y.data = 6;
z.next.data = 5;
System.out.println("x: " + x.data);
System.out.println("y: " + y.data);
System.out.println("z: " + z.data);
```

What does this code print when run?
Add the method removeEvens to LinkedIntList, which remove all even numbers from the list. You may not use an iterator or call any other LinkedIntList methods. You may declare ListNode variables, but you may not construct any new ListNode objects.

Recall the definitions of the ListNode and LinkedIntList classes:

```java
public class ListNode {
    public int data;
    public ListNode next;
}
```

```java
public class LinkedIntList {
    // Reference to the first node in the list
    private ListNode front;
    ...
}
```

Here is the declaration of the function you are to implement, as a member of the LinkedIntList class:

```java
/**
 * Removes all even numbers from this list.
 */
public void removeEvens()
```
Example Question 5

Write the method `listCompare`, which compares two lists of characters lexicographically (dictionary order). `listCompare` should return a negative integer if the first list should be ordered before the second, a positive integer if the first list should be ordered after the second, and zero if their contents are equal. Do not use any String objects or methods.

Ordering example:

```
[] < ['a', 'b', 'c'] < ['c', 'a', 't'] < ['c', 'a', 't', 's'] < ['z']
```

So your method must return a positive number if passed a first list containing `['c', 'a', 't', 's']` and a second list containing `['c', 'a', 't']`, since “cats” comes after “cat” in the dictionary. You may assume that neither list is null.