

No note-sheets, calculators, or other study aids on this exam. Write your initials at the top of each page except this one. Read through the whole exam before you get started.

Have fun!

1. (5 points total) Consider the code

```
public void displayNames(Collection<String> c) {
    c.add("Bobby");
    System.out.println("First name: "+c.get(0));
    System.out.println("All names:");
    for(String name : c) {
        System.out.println("  name: "+c);
    }
}
```

**Circle above** the **one** line in this program fails to compile when placed into IntelliJ.

2. (15 points) **Write** the Big-O runtime of the method below. **Show** your work.

```
int count = 0;
for(int i = 0; i<n; i++) {
    if(Math.random() < 0.5) {
        for(int j = 0; j<n*2; j++) {
            count ++;
        }
    } else {
        count++;
    }
}
System.out.println("count: "+count);
```

3. (15 points) **Write** the Big-O runtime for the following methods

a. In Java's ArrayList:

- i. add(E)
- ii. indexOf(Object)
- iii. contains(Object)
- iv. add(0,E)

In our ArrayList:

- v. add(E)
- vi. get(int)
- vii. remove(Object)
- viii. set(int, E)

4. (15 points) **Write** the Big-O runtime for the following methods. Assume

Iterator<String> it = list.iterator()

a. In Java's LinkedList:

- i. it.next()
- ii. size()
- iii. set(0, E)
- iv. get(n/2)

b. In our LinkedList:

- i. remove(Object)
- ii. it.hasNext()
- iii. iterator()
- iv. clear()
- v. indexOf(Object)

5. (40 points) Implement the `add(E)` method of a singly-linked `LinkedList` class that includes `head` and `size` variables in the main class.

**add**

```
public boolean add(E e)
```

Appends the specified element to the end of this list.

This method is equivalent to `addLast(E)`.

**Parameters:**

`e` - element to be appended to this list

**Returns:**

`true` (as specified by `Collection.add(E)`)

```
/** The node class, as we started it in class */
public class LinkedList<E> implements List<E> {
    private class Node {
        private Node next;
        private E value;
        private Node(Node next, E value) { /* ... */}
    }
    private Node head = null;
    // TODO: Implement add(E)
```

```
//... more of the class..
}
```

6. In this problem, you may draw memory-maps of linked lists if it helps your discussion.
- (5 points) **Explain** why, for our `LinkedList`, `add(E)` is  $O(n)$ , but for Java's it is  $O(1)$ .
  - (5 points) **Write** the Big-O runtime of `add(0,E)` for our `LinkedList`. **Explain** your answer. (This is `add(int, E)`, but only for an index of zero.)