Reminder: There is an announced, in-person quiz in lab on Friday, February 9th.

What to expect

The quiz will be closed-book, closed-notes, and no collaboration is allowed.

Expect a single sheet of paper, double-sided, with space for you to write your answers.

There will be four questions drawn from a pool including but not limited to the examples below. But, the examples below are intended to be representative – there should be no dramatic surprises on the quiz! The expectation is that you should have come into this class from 160 or other prerequisite with this level of programming and Java knowledge – if these questions or concepts seem intimidating or challenging, please come talk to one of the instructors about your preparation and how we can help make sure you're ready to succeed.

Writing code

Iteration (arrays)

Write a method with the given signature that accomplishes the stated task. Do not use any methods or classes not in the default Java scope – for example, you cannot use the utility methods of the `Arrays` class.

Tasks that require iterating through a collection and accumulating a value, such as:

- `int sum(int[] a)` - compute and return the sum of the elements in the array (return 0 if the array is of size 0)
- `int mean(int[] a)` - compute and return the mean of the elements in the array (return 0 if the array is of size 0)
- `int min(int[] a)` - return the least element of the array (return 0 if the array is of size 0))
- `int max(int[] a)` - return the greatest element of the array (return 0 if the array is of size 0))

Tasks that require iterating through a collection and testing individual items:

- `boolean allEven(int[] a)` - return true iff every element of the array is even, or if the array is of size 0 ("iff" –> "if and only if")
- `boolean someEven(int[] a)` - return true iff at least one element of the array is even, or if the array is of size 0

Tasks that require iterating through a collection and working with multiple items simultaneously:

- `boolean isSorted(int[] a)` - return true iff the array is sorted in ascending order (or if the array is of size 0)
- `boolean isDoubling(int[] a, int n)` - return true iff each element of the array at index 1 and above is `n` times larger than the previous element (or if the array is of size zero)
- `boolean isPalindrome(int[] a)` - return true iff the array is a palindrome, that is, it would be identical if reversed

Iteration (lists)

Perform any of the previous tasks, using collections of type `List`. For example, `int sum(List<Integer> l)`, and so on.

Reading and describing code

Given a method that performs a task like one of the tasks listed above under "Writing code", give a better name for the method, and succinctly describe its operation. We are *not* asking you to write a line-by-line description of what the method does; instead, we're looking for an overall description of the method. For example:

```
public static int mysteryMethod(int[] a) {
    int s = 0;
    for (int i : a) {
        s += i;
    }
    return s;
}
```

A better name for this method is something like "sum" or "sumAll" or "addAll" or the like.

A succinct description of this method that would get full credit would be something like "This method returns the sum of the values contained in the array `a`."

A description that *will not* get full credit would mechanically describe the method, for example: "The method first declares a variable s and initialized it to zero. It then iterates through the array, storing each value in a variable called `i`. It increments the value stored in `s` by `i` at each iteration. Finally, it returns the the value `s`, which is the sum of the values in the array." This answer would earn about half credit.