FOURTH QUIZ

You have 15 minutes from the start of class to complete this quiz. Give partial answers if you can’t give complete ones. Read the questions with care; work with deliberate speed. Don’t give us more than we ask for. The usual instructions apply. Good luck!

Problem 1 (6 points)

Take a look at this Java method:
```java
void flakyCode() {
    System.out.println("Hoot");
    try {
        System.out.println("Cluck");
        trouble();
        System.out.println("Gobble");
    } catch (duckException myEx) {
        System.out.println("Quack");
    } catch (Exception myEx) {
        System.out.println("Honk");
    }
    finally {
        System.out.println("Tweet");
    }
}
```

(a) What does the code above print out if the call to `trouble()` raises a `duckException`?

(b) What does the code above print out if the call to `trouble()` raises a `gooseException`?

(c) What does the code above print out if the call to `trouble()` does not raise any exceptions?

Problem 2 (see other side)
Problem 2 (14 points)

(a) (7 points) Complete the body of this method.

```java
double calcAverage (ArrayList<Integer> a) {
    // Calculate and return the average value in the argument, an ArrayList of Integers
    // (Hint: Four lines of code, not counting braces. Other variations are possible.)
    if (a.size() == 0)
        return 0;
    Integer total = 0; // 1 point for initalization. Don't count declaration.
    for (Integer i : a) { // 1 point for loop over ArrayList
        total += i; // 3 points for correctly summing values in loop (partial OK)
    }
    return total/a.size(); // 2 points for correctly computing and returning average
}
```

(b) (7 points) Complete the body of this method:

```java
Integer findLargest (ArrayList<Integer> a) {
    // Find the largest value in the argument, an ArrayList containing positive integers.
    // (Hint: Five lines of code, not counting braces. Other variations are possible.)
    Integer largestSoFar = 0;  // 1 point for initializing the largest
    for (Integer r : a)  // 1 point for loop over ArrayList
        if (r > largestSoFar) // 2 points for checking against largest so far
            largestSoFar = r;  // 2 points for assigning new maximum to largest so far correctly
    return largestSoFar;  // 1 point for returning the correct value
}
```