Computer Science I — Spring 2017
Exam #2

This exam should have four pages. Closed book and notes.

Problem 1: [24 points]

a) When should you use `equals()` instead of `==`?

b) When should variables be declared as local variables instead of instance variables? Why?

c) What is the difference between count-controlled and event-controlled loops?
public int mystery(Die d) {
    int a = 1;
    int b = 0;
    while(a != b) {
        a = b;
        b = d.roll();
        System.out.println("Rolled a "+b);
    }
    return b;
}

Problem 2: [25 points]

a) In *English*, describe what the mysterious method above does. (Try to focus on *what* it does rather than a step-by-step description of *how* it works.) To get in the right frame of mind, think about what you would write as a comment for the method.

b) Will mystery get stuck in an infinite loop if the Die instance only has 1 side? Explain.
Problem 3: [25 points]

Below, define a method called `sumRandomlySelectedValues` that takes two arguments: An array of `int`s, and a single integer specifying the number of times a value should be randomly selected from the array. It should pick values at random from the array, summing them as it goes, and return the total. It’s fine if the same value from the array is selected more than once. For full credit, it should be possible to select any of the values from the array.

For example, if passed an array containing [100,5,30] and the number 2, the method might return 130 if the two values selected at random were the first and last entries in the array. Or it might return 35 (5 plus 30), or 200 (if 100 was randomly selected twice in a row). You do not need to write an import statement as part of your solution.
Problem 4: [26 points]

The code above shows the start of the Notebook class we used in lecture and in lab. Below, define a new method, getNotesContaining, that could be added to the class. It should take a String as its input, and return an ArrayList holding all items from the notebook that contain the specified String. For example, if a Notebook instance contained “Walk the dog”, “Grade exams”, and “Feed the dog”, calling getNotesContaining("dog") would return an ArrayList containing “Walk the dog” and “Feed the dog”. (There’s a contains method in the String class that will be helpful: "doggie".contains("dog") returns true, for example.) The order of the items in the ArrayList does not matter. Your code should not modify the notes ArrayList.