Problem 1: [18 points]

a) The book says you’re not supposed to compare two objects (Strings, for example), with the == operator. Why not? How should you compare them instead?

b) In class I suggested (and Google agreed) that you should always use braces ({}’s) around the code in your if statements, even in cases where they’re not required. Why is that a good idea?

c) Why should instance variables be declared private?
public int mystery(Die d1, Die d2) {
    int x;
    int i;
    x = 0;
    i = 0;
    while (x > -10 && x < 10) {
        x = x + d1.roll();
        x = x - d2.roll();
        i = i + 1;
    }
    return i;
}

Problem 2: [20 points]

    a) In English, describe what the method 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) What impact does it have, when the method executes, if the two input Die instances don’t
        have the same number of sides?
Problem 3: [20 points]

Below, write a method called `selectString` that takes two strings as inputs and returns the “most important” of them according to the following rules: If one of the input strings starts with “Urgent”, return it. (If both of them start with “Urgent”, it doesn’t matter which you return.) If neither of them starts with “Urgent”, return the shorter of the two. If they have the same length, it doesn’t matter which you return.
Problem 4: [20 points]

Below, write a method called `getInteger` that gets an integer from the user (the keyboard) and returns it. For full credit, you should repeatedly prompt the user for an input until they’ve entered an integer that’s greater than zero. (You may assume that the Scanner class is already imported.)
Problem 5: [22 points]

In lab you worked with the BunchOfDice class that kept a list of Die instances and used them in various ways. Below, write a new method called getBigDice that could be added to BunchOfDice. The method doesn’t take any arguments, but should return an ArrayList containing all of the Die instances from the BunchOfDice with more than 6 sides. (Don’t remove them from the original list — just add them to this new list.) Assume that the list of Die instances in the BunchOfDice class is called dice, and that the ArrayList class is already imported. You might want to use the getNumSides method from the Die class as well.