PROJECT 1

Title: Dice Rolling Program
Due Date: Wednesday, September 9

Introduction

Your goal is to write (and test!) a program that rolls dice for computer games.

The software should be written as a reusable component (class). You can choose the language.

The dice roller program should allow a user (or client) to specify the number of dice to be rolled, anywhere from 1 to 5 dice at a time (for example, in Backgammon or Monopoly, two dice would be used. Risk would use up to three; Yahtzee would use up to five.) Essentially, the input is the number of dice to be rolled, and the output is the individual values of the dice.

You can assume your program only needs to roll 6-sided dice.

What you need to do:

1. Write requirements for your program. (Remember, requirements generally start with "The program shall…") The essence of many of your requirements are included the introductory paragraph above, but not necessarily all of them. Be explicit.

2. Figure out how you will test your program, and write test cases to form a comprehensive test plan. (Remember, each test case will have an initial condition, a specified input, an expected output, and an actual result.) Be sure to write your test cases in a format that Alice could both understand and execute. Avoid ambiguities such as "Run the program many times." You may have to write additional software to test your dice rolling software.

3. Write the software itself. You can some preliminary ad-hoc testing to make sure it seems to be running before you run your formal test plan.

4. For this assignment, you do not need to include test cases that check error handling, user input, etc. The scope of your test plan for this assignment is to **demonstrate that your dice roll *randomly***.

5. Run the test plan, documenting the results. NOTE: If your test plan finds an error, **do not fix your program before completing the test plan**. Instead, mark the test case as "Failed," and continue with your test plan. Once you put on your tester's hat, complete the test plan before you don your programmer's hat again. If you fix your program, rerun the entire test plan when you are done debugging.

What you should turn in:

1. The requirements (I expect this will fit on a single page).

2. The test plan, with the actual results, as if Alice had run it (I expect this will take at least a few pages). If you ran your test plan more than once, include multiple copies of the test plan (more on this later).

3. OPTIONAL: Program output from an automated test, if applicable (i.e., only if it is in some sort of tabulated or summarized or form; I do not want to see thousands of individual dice roll results on reams of paper). You can also reference printouts in the Actual Results section of each test case.

4. RECOMMENDED: Source code, if it's not too much trouble to include it. If you have more than one version of the program (because a bug was discovered by your testing), include all versions.

5. HIGHLY RECOMMEDED: A short paragraph or two summarizing what you did, and what you learned.