

CS162 Lab 3

Goals

- Implement a program according to your existing design.
- Develop a plan to test your program against requirements and verify the results.

In this lab, you will implement the program you designed in Lab 2. You will create the **Die**, **LoadedDie** and **Game** classes. You will use them in the simple **dice rolling game**.

Die class: requires an **input integer N** which determines **the number of sides on the individual die**. It includes a method to return a random integer between 1 and N as the result of rolling the die for once.

LoadedDie class: it inherits the behavior and elements of Die, but the number it returns is **biased** such that **the average output of rolling it for several times will be higher than for a Die object**. You can determine how you want to realize the biased function.

Game class: it will implement the simple dice-rolling game. In the menu function, the user will **specify the die sides used by each player** (the players can have dice with different number of sides). The user will also indicate if either or both players are **using loaded dice**, and will enter the **number of rounds in the game**. The Game class will **create the necessary objects, play the game, and display the results to the user**. The output results should **indicate the side and type** (loaded or not) **of die used** for each player, **the number of rounds won by each player** (in each round, compare the rolling results, the larger number will win), and the **final winner** of the game (the player that win more rounds).

Test the finished program. Do the results make sense? That is, when both are using the same type of dice, are the results close to 50-50? If one is using loaded dice, are the results skewed in their favor? If not, what is the problem?

What to submit:

- Your program files.
- Your makefile.
- Zip them in a single .zip file and submit it on TEACH.

Grading

- Programming style – 10%
- Correctly implement and use the Game class – 30%
- Correctly implement and use the Die class – 20%
- Correctly implement and use the LoadedDie class – 20%
- Test results using different combinations of Die and LoadedDie for each player – 10%
- Test results using different combinations of number of sides for each player – 10%