Minh Nguyen

CS 162

Lab 7: Polymorphism

Design

For this lab, after copying the same code as what was given in the instructions for the dice code, I will make a subclass called LoadedDice, under the class dice. Besides the default constructor, there will also be a constructor that takes one int parameter, which is the number of sides chosen for the dice. (In my case, I will stick with 10 throughout the program). To override the rollDice function to make the dies have a 50% chance to roll the highest possible number, I will add a random number generator that picks a number between 1 and 2. If it's 2, then it will roll the amount chosen for the number of sides on the dice, which will be the largest number possible. If it is 1, then it will simply return a random number that is belongs to each die. Then in the main function I will print out 10 rolls, calling the rollTwoDice function 10 times.

Testing

Testing for this program is quite simple. It is simply to print out the 10 different rolls to see if the loaded dice does as intended. If it works, then I should see more numbers that is the highest number possible between the sum of my two dies, since rollTwoDice will add the two dies up, in my case it should be printing out 20 more often.

Reflection

This lab was straight forward and gave me a better understanding of polymorphism and overriding virtual functions. It made me stop a while and look at how overriding the virtual function in the subclass can be called using the rollTwoDice function, and looking carefully at each parameter and which one needed to be a pointer (&). This gave me more practice with virtual functions, which is what I'll need to learn for the next lab.