

Goal/Summary: The overall goal of this program was to begin creating the game Farkle by making a basic output which included one hand of dice rolled randomly, the ability to add those dice to your meld, and the ability to score that meld based on the combos created. The program begins by showing the user the dice they rolled, with an option next to each die which allows them to move it into the meld or move a die from the meld back into their hand. Any die can be moved no matter if it creates a valid meld or not, but if any of die do create a valid meld the user has the option to either bank their points and end the game, or just end the game.

General Design: In the design I created for my program, I utilized the Die class created by Professor Crandall, and created six Die instances which I rolled and added to an Array List so that I could access them later. I then began the interaction with the user by outputting the six dice they rolled, and gave them the option to move any of the die to their meld, bank their points, or quit the game. To take in this input from the user I used a switch statement which, if the user was choosing to move a die, tested if the die was being moved to or from the meld, and performed that action. To do this, I created a meld class with a method that allowed integers to be added to its dice array field. These integers were the side that is up on the die being moved to the meld. Similarly, the side that is up could be removed from the array or returned to another class. When a die's face is added to the array it is also added to the combo array in the Combo class I created. The combo class has methods which test each test case that could produce points, and returns a score which is used in the meld class when calculating the meld score. This meld score is returned to main in Farkle and output on each screen.

Design/Programming issues: I did have some issues differentiating between the meld and combo classes, because I had too much of the computation being performed in the Combo class. I addressed this by only using the Combo class to test different meld combination cases and doing everything else in the Meld class. Additionally, I felt like main in Farkle's readability was not good, as everything looked like it was happening at once and the order it was output was not clear in my code. I addressed this by using a switch statement, separating each set of actions, and using comments to explain why different operations were being used.

Looking Back: If I were to have more time, most of what I would focus on is the efficiency of my code. A switch statement to take in the user's choice was definitely not the most efficient way to test each possibility, as much of the code is repeated, so I probably would have tried to use a for-loop instead. Additionally, I might have created a Hand class instead of handling the dice in the Farkle class so the Farkle class was mainly calling methods from instances of other classes instead of doing so many operations itself. Finally, I think I could have condensed my test cases for triplets, quadruples, quintuplets, etc. to one test case in the Combo class.

