Liar’s Dice  
Kayleigh Kinsey

I chose to create a Liar’s Dice game. Liar’s Dice was unfamiliar to me, but it seemed like it would be fun to create. Here is the description of my program from within the code itself:

This is a text-based version of the game Liar's Dice. (Rules for the game are in "Game Rules.txt") In this game, the user enters the number of players and a name for each player. The program will then roll each player's dice and the game can begin. On a player's turn, they will be shown their dice and asked to make a guess. The player must guess higher than the last player. Once the guess has been made, another player may call the first a liar. If someone chooses to call liar, the game ends. If the first player was lying, the one that called them out wins. If the first player was correct, then they win. The program will determine the winner and print a congratulations message.

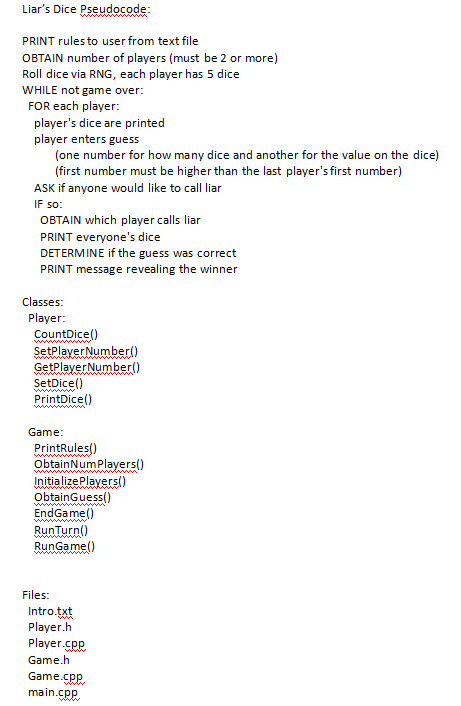
I began by doing some research into Liar’s Dice. The directions confused me at first, but I believe after reading them a few times I got a good idea of how it works. I spent some time the following week coming up with ideas. They were written out like a very rough pseudocode; a broad list of what my program would need to do. I also decided that I wanted to print out the dice to make the game more visually appealing, and spend some time sketching out how I would go about creating them. In the next few weeks, I worked on some loose pseudocode for the program. I was planning on making it more detailed, but ended up deciding that I could work well with what I had.

I started the program by creating the files I would need and setting them up by declaring the classes and methods I laid out in the pseudocode. I also declared all the variables that I knew I would need. Then, I worked through defining all the methods, beginning with the Player class because the Game class would need to use it. I also defined the runGame() method and called it in the main function so I could test out my work as I went. I completed the program in bits over the course of a few days, and ran into very few problems during the coding process. All of the issues I had were resolved quickly, and all of them were problems that I have had and solved before. The only thing that did not go well was not so much a bug as it was something that I could not figure out how to do. I planned to print out the player’s dice horizontally in a row, but I wasn’t sure how to do that without making it complicated and messy.

As I was writing this I actually thought of how I could get it to work, and though it’s a bit too late to adjust the program before turning it in, I will go and try to implement this on my own.

Once the program was mostly finished, I ran it a few times to make sure there were no bugs. There were a couple problems at first, but they were solved quickly and the debugging process did not end up taking too long. The issues that I ran into are documented in the program itself.

Here is my pseudocode:



For the most part, my final program follows the pseudocode, though there were a few methods that I needed to add that I did not think of while planning, such as a function to evaluate whether the player’s guess was right or wrong. I also had to add some methods that I originally assumed would be in runGame(), but as I was writing the code I realized it would get too long. The sequence of operations my program takes is almost the same as my pseudocode. The only exception is that I planned to print the dice of all players, but couldn’t figure out how to print all the dice horizontally to each other. It would not fit on the screen to print all the dice in a single column, so I had to cut that part out.