**Personal Experience Details:**

The project brought a lot of new experiences for me, being my first major project in Java. First, experience everything in classes was a bit of a culture shock. Even after the few weeks of learning java I found myself constantly thinking of functions to create in my main class, rather than thinking of them as methods for the class they pertained to. Java’s memory management capabilities were a nice experience; I didn’t have to spend hours worrying about what pointer I forgot to delete. I did drastically change from my original plan, which I will go into further detail in the design and implementation section, but essentially I learned it was easier to create many simple classes that only did a small job rather than fewer long and complicated classes. This turned my original plan of three classes into 28 classes.

**Design and Implementation:**

As I said earlier, I planned on only having three classes but ended with 28. I was going to have a single Card class, then create the different types of cards based on the parameters passes into the constructor.I first learned that JButtons weren’t the way to go, because I needed to access the cards which they represented. So I created my own button that contains a card. I learned that since each different type of card has a set value, name, image, cost, and for action cards its own play function, it would be easier to write and easier to follow if each type of card was its own class. These classes inherit from a parent card Class, that has their basic statistics. Certain cards are action cards, which means they can be played from the player’s hand. They all perform their own action, so they implement and interface called “ActionCard” that contains the abstract “Play” function. This also helps when looping through a hand or deck to find any card with an action by using the “insanceof.” Other cards are treasure cards, which have a value to be spent when in the player’s hand so they implement “TreasureCard.” Other cards are Victory cards containing victory points to be added up at the end of the game to determine the winner, so they implement VictoryCard. The programs main simply creates a startMenu, a class I created to hold a JFrame welcoming the user, with a start button. After the button is pressed the user enters the players’ names and players are created. A new Window is created from a class I created that takes the players as a parameter, then displays the board and the game is played within this frame. When the game ends a new frame is created with the players as parameters and a method counts the victory points in each player’s deck. The winner and the scores are then displayed in the frame.

**Concepts from class:**

* Class hierarchy (a lot of it)
* Exception handling
* System testing
* Interface programming (both my own and existing Java interfaces)
* GUI

**Future additions:**

I couldn’t think of a way to utilize text processing, but as I write this I thought I could have included a button to display the rules and those rules could be read from a text file, so I now intend to add that over this week break. I also hope to make it possible for the user to input how many players there are, so more than two people can play. Dominion also allows for the player to customize the game, by changing the action cards, so I hope to make it possible for the user to choose ten out of all the action cards in the dominion deck.