Group 8 Design Paradigm

Our group decided to create a game-box for our project 3. We chose to program in C++, and since C++ is a object oriented language, how we worked on the project very closely resembled Object-Oriented Design. To start tackling the gamebox we created class diagrams to know how we wanted to attack each game. Below is one of the first brainstorm drafts for how to implement a card game, which went through several revisions:Table

Description automatically generated

From these diagrams, we were able to determine which classes should contain what information, and appropriately assign methods to their relevant class. One of the main features of us splitting up the code into multiple classes where we could determine which information was public, private, protected. By being able to hide information, it made making mistakes in the code harder to make, as we wouldn’t be able to access certain information, like what cards are in the dealers hand. Once we had all of this information, it made implementing the features thousands of times easier than just diving straight in. One of the advantages of using object oriented design was that having multiple objects made it easier to maintain our code and by adding more functions to classes, shortened the amount of code we would leave to main.cpp. The less code in main.cpp made our code easier to follow, and simultaneously made it easier to implement new changes. Our dealershand in the deck of cards class ended up being changed from a string array to a pointer to a card array, which will help us be able to manipulate the deck and add features such as shuffling and dealing cards.