Introduction

This is my CIS-5 project that I have been continuously adding to. I hope to eventually move on from BlackJack as I feel that limiting myself to this one game is limiting how much I am learning with each project. But because the summer semester is so short I have decided to stick with it and improve and add to what I already have. I removed several parts of the game which would allowed me to focus more on the concepts I’m trying to implement and less on the old code I had already written.

Summary

*Blackjack*

The class ‘User’ contains a string that reads a name from a file (this is done from a function). It contains a few constructors used for simply creating and copying the class. There is a dealer class that holds a string that simply contains “The Dealer.” Player.h is class that is inherited from User. It is constructed by copying information from User and Dealer classes and outputs the data at the end of the game. It does this by calling a function from User. Cards.h is a class that contains the name and suit for each card. There is an array of this class for every card initialized from main. CopyClass.h is a class that simply copies the User class so that they can be incremented/decremted/added. Before the user is prompted, a file is read into the program and assigns all the unique card names into an array of classes for the program to later assign to the user and dealer. Then, a file is opened that contains a name that is then read in and assigned to a string. This string is then assigned to the player.

The first two cards of the user are generated and put into an overloaded function that is called from User.h. A random card is generated between 1-52. That random number is then stored into a parallel array of the players card array. For example, if the random number is 1, then 1 is stored in one array and Ace\_Spades is stored in an array of the player. Once the random number and unique card are stored in their respective arrays, the random number is modded to give a value between 0-13. The checkAce function is then called to see if the card that was generated was an ace. If it is, then the card is worth 11 and the bool variable ‘ace’ is set to true. Since a player can only ever have one ace in their hand that is worth 11, then all other ace cards pulled will be equal to 1. If the player does not have an ace, but will go over 21 if it is counted as 11, then the ace card is given a value of 1. Once the two cards are pulled the player then has the option to continue or fold. If he continues, the HorF function is called and will continuously loop while the player’s decision is ‘h’ or “H” and their hand is less than 21. As long as the player continues to hit, the new cards generated are added into the hand array and the values summed in a separate int variable.

Once it is the dealers turn, the Dealer class is assigned to the function that returns the data of that class. Within the function, a while loop is utilized to determine their hand. The same procedure that was utilized for the user is utilized for the dealer. Meaning the dealer’s cards and random numbers are stored in their own respective arrays. The loop ends when their hand is greater than or equal to 16.

After this, the user’s hand is sorted using a selection sort. Then a function is called that returns an array of the sorted cards in the player’s hand. Within the function, a dynamic array is created with the size of the amount of cards that are in the player’s hand. If 2 cards are in their hand then an array of size 2 is created, if there are 4 cards then an array of size 4 is created. Then, the sorted hand is printed out.

Once the game is over, several class are created that overload operators so we can increment/decrement the value of the player. Then we use another overloaded operator to add up the hands of the player and the dealer.

Once the user and dealer’s turns are over, several if-else-if statements are utilized to determine the winner. If both players exceed 21 the program outputs nobody as the winner. If either player exceeds 21, their opponent automatically wins the game. If neither player exceeded 21 then their hands are compared and whoever had the greater hand is declared the winner. Both the player and dealer’s hands are outputted into a file. In this file, the winner is declared whether it was the player or the dealer. If the player won, then the amount they won is also outputted into the file.

**Pseudo Code**

*Classes are constructed for the user and for the dealer*

*For loop that opens a file and inputs the contents of the file into the CardsInfo class array*

*Name is read from a file and assigned to a string variable that is sent to the user class*

*Do*

*Random card generated*

*Inputs that random number into an array*

*Array Hand of player is assigned a unique card*

*Random number %13 to give 0-13 value*

*checkAce function called to see if player already has an ace*

*if (rawnum==0||rawnum==11||rawnum==12||rawnum==13) //Face cards equal 10*

*player[0].hand+=10;*

*else if (rawnum==14) //first ace card is worth 11*

*player[0].hand +=11;*

*else hand+=rawnum;*

*Prints card number, unique card, and hand sum of player*

*while (cardnum1<=2)*

*If (player[0].hand ==21)*

*Skips loop, jumps to where it’s Dealers’ turn.*

*Else*

*HorF function is called that adds cards to the players hand string array while also*

*summing the values*

*checkAce function is called everytime player hits*

*function ends when hand goes over 21 or player decides to fold*

*Ace=false*

*Dealer function is called*

*While (dealers hand <16)*

*Adds card to dealers hand array while also summing cards to get dealer’s hand*

*checkAce function is called everytime to check if dealer got an ace*

*Dynamic array is created and stores the card names*

*Dynamic array is passed to Selection Sort function that sorts hand*

*Operators are overloaded to increment/decrement/add up the hands of the dealer and player*

*Function is called and prints the dynamic array/sorted hand*

*Outputs to file hand of player and hand dealer*

*if (user hand and dealer hand exceed 21) Outputs“nobody wins”*

*Else if (user exceed 21) Outputs “Dealer wins”*

*Else if (dealer exceeds 21) Outputs your name as winner*

*Else*

*Ternary operator used to analyze the winner: (Congrats, you won!) or (The Dealer won) but if both handles are the same Outputs “Tie! Nobody won!”*

*if (user won)*

*Outputs amount won in the bet*

*Input.close()*

*Out.close()*