Jose Temblador
CIS-5 (43358)
Project 2

Introduction

This version of Blackjack was incredibly time-consuming. The time spent on Project 1 and 2 combined was easily the most time I've ever invested in homework assignment. Nonetheless, I am proud of the work and effort I have put into this game. While it is not perfect, I do believe everything here functions as it should (except the bubble sort).

Total time I have invested into this game is about 50 hours of trying out different ideas, deleting entire sections of code and starting from scratch, testing, debugging, and rereading the textbook to better understand the concepts.

Summary

Blackjack

Before the user is prompted, a file is read into the program and assigns all the unique card names into an array for the program to later assign to the user and dealer. Then. A unique greeting is generated and the user is asked to enter a name and bet amount.

The first two cards of the user are generated through a do while loop. The loop ends when the player has been assigned two cards. 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 another array. 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 is 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 of 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, 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 is prompted to sort their hand using either a Selection Sort or Bubble Sort. When either function is called, the array passed into the function is the random number array, which is an int array. This is so that string compare doesn't have to utilized and so that I can use Dr. Lehr's examples that he used in class. The only problem that I was not able to solve was an issue with the Bubble Sort. The Bubble Sort function only works if there are more than 4 elements in the random number array. If the Bubble Sort function is called with only 2 or 3 elements in the array then the program ends. The Selection Sort function works regardless of how many elements are in the array.

The user is then asked if they would like to search for any duplicate cards in the player or dealer's hand. Since there is no checking when the cards are pulled, the same cards may show up in both the player and dealer's hand. The user is given the option to skip the search. If they choose to search, then a Linear Search function is called and will return the card that was found in both hands. If no matching card is found then the "No match was found" is displayed.

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.

Jose Temblador
CIS-5 (43358)
Project 2

Pseudo Code

```
Initialize
For loop that opens a file and inputs the contents of the file into an array
Random switch case greeting is chosen Asks for user name
Input bet amount
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
      hand+=10;
   else if (rawnum==14)   //first ace card is worth 11
      hand+=11;
   else hand+=rawnum;
    Prints card number, unique card, and hand sum of player
while (cardnum1<=2)
If (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
Displays current hand
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
Displays dealers hand
Switch case prompting user to choose between sorting with Binary or Selection Sort
Sorts player's hand array of cards based off what player chooses
Switch case prompting user between searching both hands to find duplicate cards (Linear
search)
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()
```