**Project 1**

**Blackjack**

**CSC-17A-48290**

**Cameron Hartling**

**11/5/2021**

**Introduction**

Title: Blackjack

This is a Blackjack game that has the ability to save the player’s initials and wins for the session. The rules of Blackjack are standard. The standard deck of fifty-two cards is shuffled then two cards are dealt each to the dealer and player. The player is then given the chance to get another card or pass to the dealers turn. The goal of the game is to be the closest to 21 without going over. The points for the cards are 11 for ace; 10 for the king, queen, jack, and ten; and the rest are face value.

Once the game is finished the player has the option to replace their data with the previous data or to start another game.

**Summary**

Project size: about 315 lines

The number of variables: about 28

This project includes many concepts from the ninth through twelfth from the Gaddis textbook.

It was a good amalgamation of the concepts covered so far in a single program. I was having trouble with the binary file and Netbeans was showing that it could not resolve the identifier .write() but it would compile properly. This took a little over a week to finish the final project with the majority taken in the third iteration of the program. The usage of structures in C++ is a tremendous space and time saver in how a program is laid out. In version 1 of my project, I had used arrays to hold the deck and the card values which took an extra function to determine the values of the cards. With a structure I was able to link the card name and the card’s value together.

To make this program function better I believe that the use of classes would help. Some visual elements could be added to improve this program as well.

**Flow Chart**

See: HartlingCameronPProj1FlowChart.xml

**Pseudo Code**

// Function Prototypes

// void CreateDeck(string\* deck);

// void Shuffle(string\* deck);

// int CardVal(string card);

// void Blackjack();

//Main function

//Initialize Variables

// Explain the rules of Blackjack

//Ask user for name

//Get users initials

//Welcome message

//While loop for asking user to play blackjack

// Ask user if they want to play

// If yes call Blackjack function

// Thank you message for playing

// Function to create a deck

// void CreateDeck(string\* deck)

// Initialize variables

// Two string arrays containing card names and suits

//Populate deck

// For loop for the card names

// For loop for suit

// Add card name and suit together and put in deck[]

// Function to shuffle array of cards

// void Shuffle(string\* deck)

// set rand seed

// Initialize variables

//Populate deck by calling CreateDeck()

//Randomly shuffle the deck using for loop and swapping random places

// Function for card value

// int CardVal(string card)

//Initialize

// string array for card name

// int array for values of card names

// For loop that goes through and checks for equivalent card name in array and string parameter

// return value in the same position as the int array

// Blackjack gameplay

// void Blackjack()

//Initialize variables

// Create dynamic arrays for deck, dealer, and player

//Get a shuffled deck by calling Shuffle(deck)

//Deal the first two cards by moving cards from deck to dealer and player

// Add values of the cards to dealerTot and playerTot

// Check to see if dealer has a Blackjack

// else

// While loop for getting more cards for user

//Ask user if they want to hold or get another card

// Print out users hand, player[]

// Print out the total hand value

// Ask if they want another card

//Add new card if chosen

//If over 21 print losing statement

//Else If exactly 21 print winning statement

//else

// Print dealers’ hand and total

//Add new cards to the dealer until they win or go over

// If dealerTot is over 21 print winning statement for player

// else print losing statement for player

**Concepts**

|  |  |
| --- | --- |
| **Chapter** | **Concept** |
|  |  |
| **9** | **Pointer Variables** |
|  | **Arrays/Pointers** |
|  | **Function Parameters** |
|  | **Memory Allocation** |
|  | **Return Parameters** |
|  |  |
| **10** | **C-strings** |
|  | **Strings** |
|  |  |
| **11** | **Arrays** |
|  | **Nested** |
|  | **Function Arguments** |
|  | **Function Return** |
|  | **Pointers** |
|  | **Enumeration** |
|  |  |
| **12** | **Formatting** |
|  | **Random Access Files** |
|  | **Binary Files** |
|  | **Input/Output Simultaneous** |
|  | **Member Functions** |

**References**

1. “Starting Out with C++ from Control Structures to Objects” by Tony Gaddis