**Project 2**

Title

**Blackjack**

Course

**CSC 5**

Section

**42450**

Due Date

**June 9, 2014**

Author

**Victor Medel**

**Introduction**

Title: Blackjack

This version continues to be a simple program that allows any player to quickly play a game of Blackjack. The program starts up with a menu with three options, *1.* *Play Blackjack*, *2.* *Blackjack Game Overview,* or *Anything Else to Exit*. The options are very straight forward, option one allows the user to play the game, option two provide the quick overview of the game as written below, and any other input exits the user from program.

The object of the game is to beat the house by receiving a score of 21 or by getting a higher score than the house without going over 21 with any additional cards. The game begins by dealing two cards to the player; after displaying your score and if your score is less than 21 you will have the option to take another card to add to your total score or hold with your existing score. If you hold or go over 21 after choosing another card the program will automatically display the house’s hand and then determine the outcome. Multiple decks of cards are used with the following values:

Cards 2 through 10 = face value points

Jacks = 10 points

Queens = 10 Points

Kings = 10 Points

Aces = 1 or 11 depends on the player’s total. If player’s total is less than ten points then they

hold a value of eleven otherwise the card will hold a value of one.

**Summary**

Developing the program took over two weeks and several versions due my continued limited experience with C++ programing, speed of the class lectures , and just like Project 1, the project packet development. As references I used Project 1, all class lecture examples posted on GitHub, the course textbook (*Problem Solving with C++ by Walter Savitch*), the web to obtain some of the rules on how to play Blackjack as well as how to start this project. I also utilized the sample project documentation provided on Black Board to help me with the production of this document.

I’ve developed this program utilizing many of the concepts that have been covered by the class textbook (*Problem Solving with C++ 8th Edition by Walter Savitch*) within chapters one through seven. I have also used concepts discussed during class lecture and lab to create this program. The program runs as expected but I believe that this program still has many opportunities. The lack of time left in this semester has limited my ability to continue to learn C++ and really improve this program. I would like to believe that I would continue to read the class textbook over the summer but the reality is that I will need to enroll in the next level of C++ programming classes to continue to learn, which I am planning on doing.

One of the major obstacles that I encountered while developing this program was the ability to utilize a two dimensional array to deal a card; therefore, I developed the program to work with a one dimensional array only. I did include the code that I was attempting to work at the very end of my code, lines 361-392.

I did try to include all of the concepts covered in class but due to the limited time to develop this program I mainly concentrated on the following:

1. Functions and Arrays [Lines 313-319]
2. One and Two Dimensional Arrays [Lines 313-319]
3. Passing Arrays Between Functions [Lines 313-326]
4. Pass by Value [Line 359]
5. Pass by Reference [Lines 313-326]
6. Defaulted Parameters [Lines 21-23]
7. Returning Primitive Data Types [Lines 359]
8. Formatting [Lines 323-325]
9. Reading and Writing to Files [Lines 59-67 & 69-78]

**Concepts Used**

From Textbook:

*Problem Solving with C++ 8th Edition by Walter Savitch*

Chapter 2

2.1 Variables and Assignments

2.2 Input and Output

2.3 Data Types and Expressions

2.4 Simple Flow Control

2.5 Program Style

Chapter 3

3.1 Using Boolean Expressions

3.2 Multiway Branches

3.3 More About C++ Loop Statements

3.4 Designing Loops

Chapter 4

4.1 Top-Down Design

4.2 Predefined Functions

4.3 Programmer-Defined Functions

4.4 Procedural Absraction

4.5 Scope And Local Variables

Chapter 5

5.1 void Functions

5.2 Call-By-Reference Parameters

~~5.3 Using Procedural Abstraction~~

5.4 Testing and Debugging Functions

5.5 General Debugging Techniques

Chapter 6

6.1 Streams and Basic File I/O

6.2 Tools for Stream I/O

6.3 Character I/O

Chapter 7

7.1 Introduction to Arrays

7.2 Arrays in Functions

7.3 Programming with Arrays

7.4 Multidimensional Array

From Class Lectures and Lab:

1. Input and Output
2. Loops
3. Menus
4. Branching Constructs
5. Mathematical Expressions
6. User interactivity
7. Boolean Expressions
8. Functions
9. One Dimensional Arrays
10. Two Dimensional Arrays
11. Void Functions
12. Programming Logic

**Code Specifications**

|  |  |
| --- | --- |
| Lines of Code |  |
| Comment Lines |  |
| Blank Lines |  |
| Total Lines of Source Files | **392** |
| Number of Variables |  |

**Variables Used**

|  |  |  |
| --- | --- | --- |
| Type | Variable Name | Description |
| Integer | valu | Function parameter that hold the dealt card value within the main program |
|  | pTotal | Holds the players total score within the main program. It is also used as a function parameter that holds the players total score within the function definition and function header |
|  | rTotal | Utilized to keep a running total of the player score |
|  | total | Function parameter used to hold a card value total |
|  | hTot | Holds the house’s total score |
|  | hrTot | Utilized to keep a running total of the house’s score |
|  | choice | Menu selection input |
|  | randCard | Randomly selects a number/card from 1 to 14 |
|  | randSuit |  |
|  |  | Randomly selects a number/suit from 1 to 4 |
| String | card[] | Array that outputs a card value |
|  | suit[] | Array that outputs a suit value |
|  | next | Variable used to read from a file |
|  | name | Variable used to write a name to a file |
|  |  |  |
| Ifstream | infile | Variable used to identify file that is being written to |
|  |  |  |
| Ofstream | outfile | Variable used to identify file that is being read from |
|  | outfile2 | Variable used to identify file that is being read from |
|  |  |  |
| Character | ans | Input option to allow continue of play |
|  |  |  |
| Boolean | exitMnu | Alternative option to end program at menu selection |
|  |  |  |

**Flowchart**

**Program Code**