**Project 1**

Title

**Simple Blackjack Game**

Course

**CSC 11**

Section

**48598**

Due Date

**November 3, 2014**

Author

**Victor Medel**

**Introduction**

Title: Blackjack

This is a simple program that allows any player to quickly play a game of Blackjack. 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 the additional 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 are determined by the random number generated

**Summary**

Developing the program took about two weeks and several versions due my limited experience with C++ programing and the project packet development. As references I used the course textbook (*Problem Solving with C++ by Walter Savitch*) and 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 four and partially five. 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. As I continue to cover new material in the class I believe I can improve this program and raise it to a more sophisticated programming level.

One of the major obstacles that I encountered while developing this program was the ability to pull an individual value from a function. Reading chapter five in the textbook I was able to find the **call-by-reference parameter**, which allows you to do just that. The **call-by-reference parameter** substitutes the function argument output for formal parameter of the function. This allowed me to pull the card value from the function making it easier to keep a running total score.

**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

~~4.6 Overloading Function Names~~

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~~

From Class Lectures and Lab:

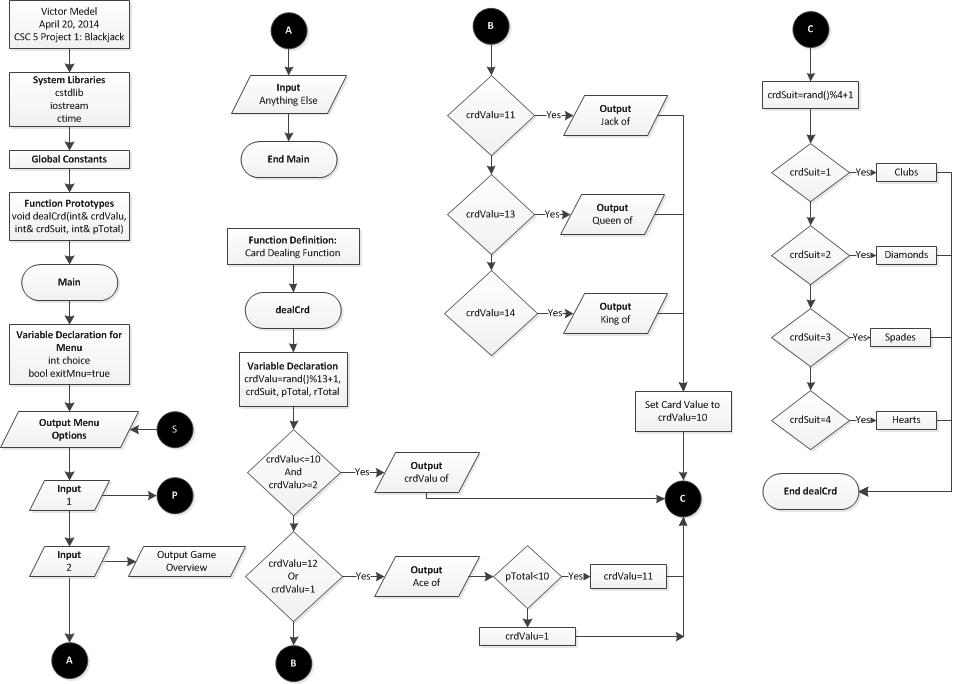
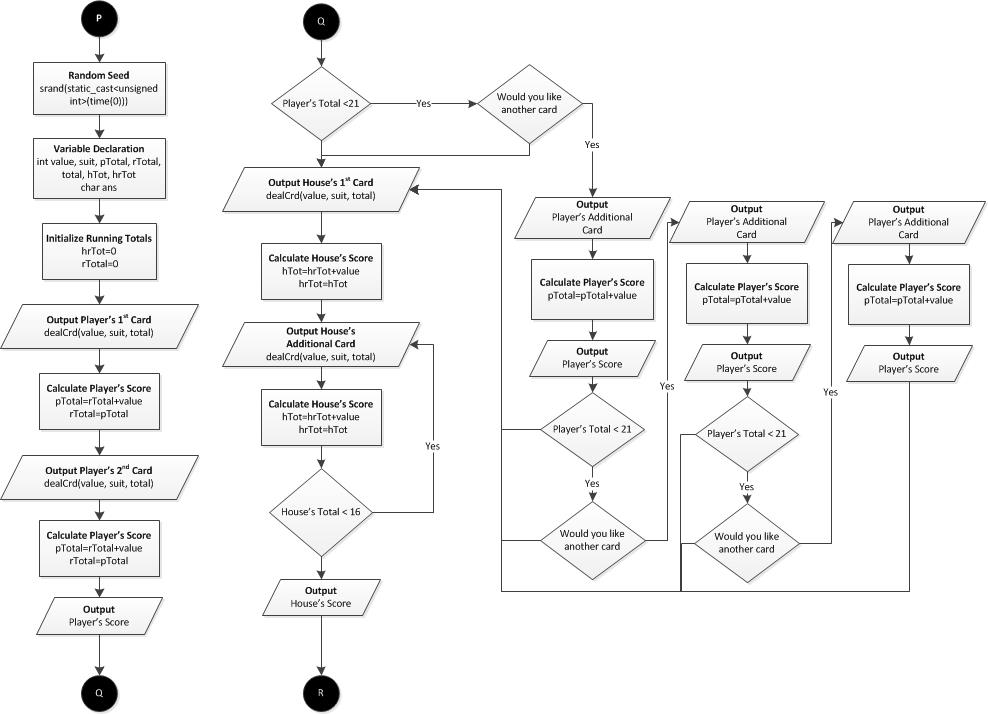
1. Input and Output
2. Loops
3. Menus
4. Branching Constructs
5. Mathematical Expressions
6. User interactivity
7. Boolean Expressions
8. Functions

**Code Specifications**

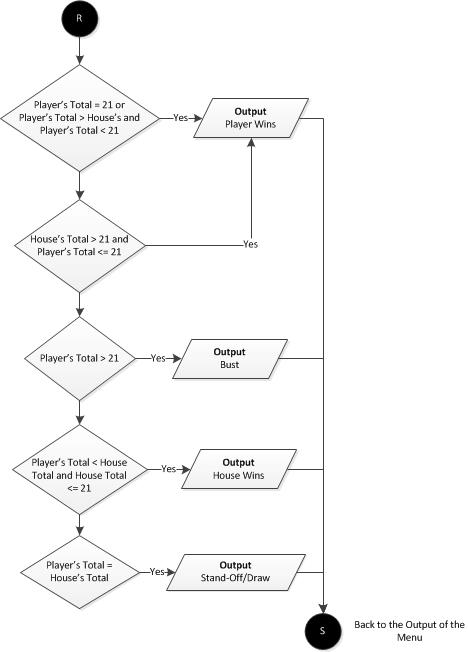
|  |  |
| --- | --- |
| Lines of Code | 172 |
| Comment Lines | **32** |
| Blank Lines | **4** |
| Total Lines of Source Files | **208** |
| Number of Variables | **12** |

**Variables Used**

|  |  |  |
| --- | --- | --- |
| Type | Variable Name | Description |
| Integer | value | Function parameter that hold the dealt card value within the main program |
|  | suit | Function parameter that holds the suit of the card dealt 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 |
|  | crdValu | Function parameter that holds the card value within the function definition and function header |
|  | crdSuit | Function parameter that holds the card value within the function definition and function header |
|  |  |  |
| Character | ans | Input option to allow continue of play |
|  |  |  |
| Boolean | exitMnu | Alternative option to end program at menu selection |
|  |  |  |
|  |  |  |

****

**Flowchart**

****

**Program Code**

1 /\*

2 \* File: main.cpp

3 \* Author: Victor Medel

4 \* Created on April 15, 2014, 9:55 AM

5 \* CSC 5 (42450) | Project 1: Black Jack Game

6 \*/

7

8 //System Libraries

9 #include <cstdlib>//Random function srand

10 #include <iostream>//Standard input/output

11 #include <ctime>//time for random and program

12 using namespace std;

13

14 //Global Constants

15

16 //Function Prototypes

17 void **dealCrd**(int& crdValu, int& crdSuit, int& pTotal);

18

19 //Execution Starts Here

20 int **main**(int argc, char\*\* argv) {

21 //Declare menu variables

22 int choice;

23 bool exitMnu=true;

24 //Loop until exit

25 do{

26 //Output Menu

27 cout<<**"\n"**;

28 cout<<**"**Select From The Menu**"**<<endl;

29 cout<<**"\n"**;

30 cout<<**"**1. Play Blackjack**"**<<endl;

31 cout<<**"**2. Blackjack Game Overview**"**<<endl;

32 cout<<**"**\*\*\*Anything Else Exit Program\*\*\***"**<<endl;

33 cout<<**"\n"**;

34 //Input your choice

35 cout<<**"**Selection: **"**;

36 cin>>choice;

37 //Solve the problem chosen

38 switch(choice){

39 case 1:

40 //Random Seed and Variable Declaration

41 srand(static\_cast<unsigned int>(time(0)));

42 int value, suit, pTotal, rTotal, total, hTot, hrTot;

43 char ans;

44 hrTot=0;

45 rTotal=0;

46 //Player's Initial Hand

47 cout<<**"**You have been dealt the following cards: **"**;

48 cout<<**"\n"**;

49 dealCrd(value, suit, total);

50 pTotal=rTotal+value;

51 rTotal=pTotal;

52 cout<<**"** | **"**;

53 dealCrd(value, suit, total);

54 pTotal=rTotal+value;

55 cout<<**"\n"**;

56 cout<<**"\n"**;

57 cout<<**"**Your score is: **"**;

58 cout<<pTotal;

59 cout<<**"\n\n"**;

60 //Option to Allow Player to Hit and Continue Playing

61 //Three additional cards always exceed a score of 21

62 if(pTotal<21){

63 cout<<**"**Would you like another card?**\n"**;

64 cout<<**"**Enter y for yes, anything else for no: **"**;

65 cin>>ans;

66 if (ans=='y'||ans=='Y'){

67 //Players Additional Cards

68 cout<<**"\n"**;

69 cout<<**"**You have been dealt a **"**;

70 dealCrd(value, suit, total);

71 pTotal=pTotal+value;

72 cout<<**"\n"**;

73 cout<<**"**Your score is now: **"**;

74 cout<<pTotal;

75 cout<<**"\n\n"**;

76 if (pTotal<21){

77 cout<<**"**Would you like another card?**\n"**;

78 cout<<**"**Enter y for yes, anything else for no: **"**;

79 cin>>ans;

80 if (ans=='y'||ans=='Y'){

81 //Players Additional Card

82 cout<<**"\n"**;

83 cout<<**"**You have been dealt a **"**;

84 dealCrd(value, suit, total);

85 pTotal=pTotal+value;

86 cout<<**"\n"**;

87 cout<<**"**Your score is now: **"**;

88 cout<<pTotal;

89 cout<<**"\n\n"**;

90 if (pTotal<21){

91 cout<<**"**Would you like another card?**\n"**;

92 cout<<**"**Enter y for yes, anything else for no: **"**;

93 cin>>ans;

94 if (ans=='y'||ans=='Y'){

95 //Players Additional Card

96 cout<<**"\n"**;

97 cout<<**"**You have been dealt a **"**;

98 dealCrd(value, suit, total);

99 pTotal=pTotal+value;

100 cout<<**"\n"**;

101 cout<<**"**Your score is now: **"**;

102 cout<<pTotal;

103 cout<<**"\n\n"**;

104 }

105 }

106 }

107 }

108 }else;

109 }

110 //House's Hand

111 cout<<**"\n"**;

112 cout<<**"**The house has been dealt the following cards: **"**;

113 cout<<**"\n"**;

114 dealCrd(value, suit, total);

115 hTot=hrTot+value;

116 hrTot=hTot;

117 do{

118 cout<<**"** | **"**;

119 dealCrd(value, suit, total);

120 hTot=hrTot+value;

121 hrTot=hTot;

122 //Based on Blackjack Rules House continues to deal

123 //itself a card if total score is less than 16

124 }while(hTot<16);

125 cout<<**"\n"**;

126 cout<<**"**The house's score is: **"**;

127 cout<<hTot;

128 cout<<**"\n"**;

129 //Outcome Output

130 if(pTotal==21||(pTotal>hTot&&pTotal<21)){

131 cout<<**"\n"**;

132 cout<<**"**Congratulations! You have won**"**;

133 cout<<**"\n"**;

134 }else if(hTot>21&&pTotal<=21) {

135 cout<<**"\n"**;

136 cout<<**"**Congratulations! You have won**"**;

137 cout<<**"\n"**;

138 }else if(pTotal>21){

139 cout<<**"\n"**;

140 cout<<**"**Bust**"**;

141 cout<<**"\n"**;

142 }else if(pTotal<hTot&&hTot<=21){

143 cout<<**"\n"**;

144 cout<<**"**House Wins**"**;

145 cout<<**"\n"**;

146 }else if(pTotal==hTot){

147 cout<<**"\n"**;

148 cout<<**"**Stand-Off/Draw, Play Again**"**;

149 cout<<**"\n"**;

150 }

151 //Exit Stage Right

152 break;

153 case 2:

154 cout<<**"\n"**;

155 cout<<**"**The object of the game is to beat the house **\n"**;

156 cout<<**"**by receiving a score of 21 or by getting a higher **\n"**;

157 cout<<**"**score than the house without going over 21 with**\n"**;

158 cout<<**"**any additional cards. The game begins by dealing**\n"**;

159 cout<<**"**two cards to the player; after displaying your**\n"**;

160 cout<<**"**score you will have the option to take another**\n"**;

161 cout<<**"**card to add to your total score or hold with**\n"**;

162 cout<<**"**your existing score. If you hold or go over 21**\n"**;

163 cout<<**"**after choosing another card the program will **\n"**;

164 cout<<**"**automatically display the house's hand and then**\n"**;

165 cout<<**"**determine the outcome.**\n"**;

166 ;break;

167 default: exitMnu=false;

168 }

169 }while(exitMnu);

170 //Exit Stage Right

171 return 0;

172 }

173 //Function Definition (Card Dealing Function)

174 void **dealCrd**(int& crdValu, int& crdSuit, int& pTotal){

175 //Randomly selects card values

176 int rTotal;

177 crdValu=rand()%14+1;

178 if (crdValu<=10&&crdValu>=2){

179 cout<<crdValu;

180 cout<<**"** of **"**;

181 }if (crdValu==11){

182 cout<<**"**Jack of **"**;

183 }if (crdValu==12||crdValu==1){

184 cout<<**"**Ace of **"**;

185 if (pTotal<10){

186 crdValu=11;

187 }else

188 crdValu=1;

189 }if (crdValu==13){

190 cout<<**"**Queen of **"**;

191 crdValu=10;

192 }if (crdValu==14){

193 cout<<**"**King of **"**;

194 crdValu=10;

195 }

196 //End of card value selection

197 //Randomly selects suit

198 crdSuit=rand()%4+1;

199 if(crdSuit==1){

200 cout<<**"**Clubs**"**;

201 }if(crdSuit==2){

202 cout<<**"**Diamonds**"**;

203 }if(crdSuit==3){

204 cout<<**"**Spades**"**;

205 }if (crdSuit==4){

206 cout<<**"**Hearts**"**;

207 }

208 //End of Suit Selection

209 }