**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 fully understand the final chapters that have been covered in this course. 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. My plan is to fully understand Arrays by the end of June and as far as the rest of the material, I will wait for CSC 17A.

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.

The course covered many basic concepts of C++ programing and 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 items listed on the assignment prompt which are the following:

1. Functions and Arrays [Lines 313-319]
2. One and Two Dimensional Arrays [Lines 30 & 31, 2D attempt on 364]
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 | 303 |
| Comment Lines | **84** |
| Blank Lines | **4** |
| Total Lines of Source Files | **391** |
| Number of Variables | **18** |

**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**

1 /\*

2 \* File: main.cpp

3 \* Author: Victor Medel

4 \* Created on June 1, 2014, 9:30 PM

5 \* CSC 5 (42450) | Project 2: Black Jack Game Version 2

6 \*/

7

8 //System Libraries

9 #include <iostream>

10 #include <ctime>

11 #include <cstdlib>

12 #include <string>

13 #include <iomanip>

14 #include <fstream>

15 using namespace std;

16

17 //Global Constants

18

19 //Function Prototypes

20 void **dealCrd**(string card[], int& randCard, string suit[], int& randSuit);

21 void **prntCrd**(string card[], int& randCard, string suit[], int& randSuit);

22 int **calc**(int& randCard, int& valu);

23

24 //Execution Starts Here

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

26 //Variable Declaration and Random Seed

27 srand(time(0));

28 int valu, pTotal, rTotal, hTotal, hrTot;

29 int randCard, randSuit;

30 string suit[4]={**"**of Hearts**"**,**"**of Diamonds**"**,**"**of Spades**"**,**"**of Clubs**"**};

31 string card[14]={**"**Ace**"**,**"**2**"**,**"**3**"**,**"**4**"**,**"**5**"**,**"**6**"**,**"**7**"**,**"**8**"**,**"**9**"**,**"**10**"**,**"**Ace**"**,**"**Jack**"**,**"**Queen**"**,**"**King**"**};

32 char ans;

33 ifstream infile;

34 ofstream outfile,outfile2;

35 int choice;

36 bool exitMnu=true;

37 string next,name;

38 //Loop until exit

39 do{

40 //Output Menu

41 cout<<**"\n"**;

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

43 cout<<**"\n"**;

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

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

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

47 cout<<**"\n"**;

48 //Input your choice

49 cout<<**"**Selection: **"**;

50 cin>>choice;

51 //Solve the problem chosen

52 switch(choice){

53 case 1:

54 //Initialize Player and House Running Totals

55 hrTot=0;

56 rTotal=0;

57 //Open File to write name

58 outfile2.open(**"**output2.dat**"**, ios::app);

59 if(outfile2.fail()){

60 cout<<**"**Input file failed to open.**\n"**;

61 exit(1);

62 }

63 cout<<**"**Please enter your first name: **"**;

64 cin>>name;

65 outfile2<<name<<**"** **"**;

66 outfile2.close();

67 //Open File to read welcome message

68 infile.open(**"**input.dat**"**);

69 if(infile.fail()){

70 cout<<**"**Input file failed to open.**\n"**;

71 exit(1);

72 }

73 cout<<name<<**"** **"**;

74 while (infile>>next){

75 cout<<next<<**"** **"**;

76 }

77 infile.close();

78 cout<<endl<<endl;

79 //Function Call

80 dealCrd(card,randCard,suit,randSuit);

81 prntCrd(card,randCard,suit,randSuit);

82 calc(randCard, valu);

83 //Ace Re-Adjustment

84 if(valu==1){

85 valu=11;

86 }

87 //End of Ace Re-Adjustment

88 pTotal=rTotal+valu;

89 rTotal=pTotal;

90 cout<<**"** | **"**;

91 dealCrd(card,randCard,suit,randSuit);

92 prntCrd(card,randCard,suit,randSuit);

93 calc(randCard, valu);

94 //Ace Re-Adjustment for Additional Cards

95 if(rTotal>10&&valu==11){

96 valu=1;

97 }else if(rTotal<=10&&valu==1){

98 valu=11;

99 }

100 //End of Ace Re-Adjustment

101 pTotal=rTotal+valu;

102 cout<<**"\n\n"**;

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

104 cout<<pTotal;

105 cout<<**"\n"**;

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

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

108 if(pTotal<21){

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

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

111 cin>>ans;

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

113 //Players Additional Cards

114 cout<<**"\n"**;

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

116 dealCrd(card,randCard,suit,randSuit);

117 prntCrd(card,randCard,suit,randSuit);

118 calc(randCard, valu);

119 //Ace Re-Adjustment for Additional Cards

120 if(pTotal>10&&valu==11){

121 valu=1;

122 }else if(pTotal<=10&&valu==1){

123 valu=11;

124 }

125 //End of Ace Re-Adjustment

126 pTotal=pTotal+valu;

127 cout<<**"\n"**;

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

129 cout<<pTotal;

130 cout<<**"\n\n"**;

131 if (pTotal<21){

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

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

134 cin>>ans;

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

136 //Players Additional Card

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

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

139 dealCrd(card,randCard,suit,randSuit);

140 prntCrd(card,randCard,suit,randSuit);

141 calc(randCard, valu);

142 //Ace Re-Adjustment for Additional Cards

143 if(pTotal>10&&valu==11){

144 valu=1;

145 }else if(pTotal<=10&&valu==1){

146 valu=11;

147 }

148 //End of Ace Re-Adjustment

149 pTotal=pTotal+valu;

150 cout<<**"\n"**;

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

152 cout<<pTotal;

153 cout<<**"\n\n"**;

154 if (pTotal<21){

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

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

157 cin>>ans;

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

159 //Players Additional Card

160 cout<<**"\n"**;

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

162 dealCrd(card,randCard,suit,randSuit);

163 prntCrd(card,randCard,suit,randSuit);

164 calc(randCard, valu);

165 //Ace Re-Adjustment for Additional Cards

166 if(pTotal>10&&valu==11){

167 valu=1;

168 }else if(pTotal<=10&&valu==1){

169 valu=11;

170 }

171 //End of Ace Re-Adjustment

172 pTotal=pTotal+valu;

173 cout<<**"\n"**;

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

175 cout<<pTotal;

176 cout<<**"\n\n"**;

177 //Fourth Card Option for the Risk Taker

178 if(pTotal<21){

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

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

181 cin>>ans;

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

183 //Players Additional Card

184 cout<<**"\n"**;

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

186 dealCrd(card,randCard,suit,randSuit);

187 prntCrd(card,randCard,suit,randSuit);

188 calc(randCard, valu);

189 //Ace Re-Adjustment for Additional Cards

190 if(pTotal>10&&valu==11){

191 valu=1;

192 }else if(pTotal<=10&&valu==1){

193 valu=11;

194 }

195 //End of Ace Re-Adjustment

196 pTotal=pTotal+valu;

197 cout<<**"\n"**;

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

199 cout<<pTotal;

200 cout<<**"\n\n"**;

201 }

202 }

203 }

204 }

205 }

206 }

207 }else;

208 }

209

210 //House's Hand

211 cout<<**"\n"**;

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

213 cout<<**"\n"**;

214 dealCrd(card,randCard,suit,randSuit);

215 prntCrd(card,randCard,suit,randSuit);

216 calc(randCard, valu);

217 //Ace Re-Adjustment for Additional Cards

218 if(hTotal>10&&valu==11){

219 valu=1;

220 }else if(hTotal<=10&&valu==1){

221 valu=11;

222 }

223 //End of Ace Re-Adjustment

224 hTotal=hrTot+valu;

225 hrTot=hTotal;

226 do{

227 cout<<**"** | **"**;

228 dealCrd(card,randCard,suit,randSuit);

229 prntCrd(card,randCard,suit,randSuit);

230 calc(randCard, valu);

231 //Ace Re-Adjustment for Additional Cards

232 if(hTotal>10&&valu==11){

233 valu=1;

234 }else if(hTotal<=10&&valu==1){

235 valu=11;

236 }

237 //End of Ace Re-Adjustment

238 hTotal=hTotal+valu;

239 //Based on Blackjack Rules House continues to deal

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

241 }while(hTotal<16);

242 cout<<**"\n"**;

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

244 cout<<hTotal;

245 cout<<**"\n"**;

246 //Outcome Output

247 //Open File to write results

248 outfile.open(**"**output.dat**"**, ios::app);

249 if(outfile.fail()){

250 cout<<**"**Input file failed to open.**\n"**;

251 exit(1);

252 }

253 if(pTotal==21||(pTotal>hTotal&&pTotal<21)){

254 cout<<**"\n"**;

255 cout<<**"**\*\*\*Congratulations! You have won\*\*\***"**;

256 outfile<<**"**Win**"**<<**"** **"**<<endl;

257 cout<<**"\n"**;

258 }else if(hTotal>21&&pTotal<=21) {

259 cout<<**"\n"**;

260 cout<<**"**\*\*\*Congratulations! You have won\*\*\***"**;

261 outfile<<**"**Win**"**<<**"** **"**<<endl;

262 cout<<**"\n"**;

263 }else if(pTotal>21){

264 cout<<**"\n"**;

265 cout<<**"**\*\*\*Bust\*\*\***"**;

266 outfile<<**"**Loss**"**<<**"** **"**<<endl;

267 cout<<**"\n"**;

268 }else if(pTotal<hTotal&&hTotal<=21){

269 cout<<**"\n"**;

270 cout<<**"**\*\*\*House Wins\*\*\***"**;

271 outfile<<**"**Loss**"**<<**"** **"**<<endl;

272 cout<<**"\n"**;

273 }else if(pTotal==hTotal){

274 cout<<**"\n"**;

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

276 outfile<<**"**Tie**"**<<**"** **"**<<endl;

277 cout<<**"\n"**;

278 }

279 outfile.close();

280 //Exit Stage Right

281 break;

282 case 2:

283 //Objective of my game

284 cout<<**"\n"**;

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

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

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

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

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

290 cout<<**"**score and if your score is less than 21 you **\n"**;

291 cout<<**"**will have the option to take another**\n"**;

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

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

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

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

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

297 cout<<**"**Multiple decks of cards are used with the following values:**\n\n"**;

298 cout<<**"**Cards 2 through 10 = face value points**\n"**;

299 cout<<**"**Jacks = 10 points**\n"**;

300 cout<<**"**Queens = 10 Points**\n"**;

301 cout<<**"**Kings = 10 Points**\n"**;

302 cout<<**"**Aces = 1 or 11 depends on the player's total. If player's total **\n"**;

303 cout<<**"**is less than ten points then they hold a value of eleven otherwise **\n"**;

304 cout<<**"**the card will hold a value of one.**\n"**;

305 ;break;

306 default: exitMnu=false;

307 }

308 }while(exitMnu);

309 //Exit Stage Right

310 return 0;

311 }

312 void **dealCrd**(string card[], int& randCard, string suit[], int& randSuit){

313 //Randomly Selects Card and Suit

314 randSuit=rand()%4;

315 randCard=rand()%14;

316 card[randCard];

317 suit[randSuit];

318 }

319 //Prints Card

320 void **prntCrd**(string card[], int& randCard, string suit[], int& randSuit){

321 //Outputs One Card

322 cout<<right<<setw(2)<<card[randCard];

323 cout<<**"** **"**;

324 cout<<left<<setw(2)<<suit[randSuit];

325 }

326 //Function Calculates the Value of each card

327 int **calc**(int& randCard, int& valu){

328 //Assigns Value to Card Dealt

329 if(randCard==0){

330 valu=1;

331 }else if(randCard==1){

332 valu=2;

333 }else if(randCard==2){

334 valu=3;

335 }else if(randCard==3){

336 valu=4;

337 }else if(randCard==4){

338 valu=5;

339 }else if(randCard==5){

340 valu=6;

341 }else if(randCard==6){

342 valu=7;

343 }else if(randCard==7){

344 valu=8;

345 }else if(randCard==8){

346 valu=9;

347 }else if(randCard==9){

348 valu=10;

349 }else if(randCard==10){

350 valu=11;

351 }else if(randCard==11){

352 valu=10;

353 }else if(randCard==12){

354 valu=10;

355 }else if(randCard==13){

356 valu=10;

357 }

358 return valu;

359 }

360 /\*

361 \* I attempted to use 2D array but it was not functional with the Blackjack

362 \* game I was developing. I've included the code to demonstrated my attempt.

363 \*

364 void dealCrd(int card[][COLS], int n, int randSuit){

365 for(int i=0;i**<n;i++){**

366 **card[i][0]=rand()%14+1;**

367 **card[i][1]=suit(randSuit);**

368 **}**

369 **}**

370

371 **void prntCrd(int card[][COLS], int n, int randSuit){**

372 **for (int i=0;i<n;i++){**

373 **cout<<card[i][0];**

374 **cout<<" of ";**

375 **if(card[i][0]==0){**

376 **cout<<"Hearts";**

377 **}else if(card[i][0]==1){**

378 **cout<<"Diamonds";**

379 **}else if(card[i][0]==2){**

380 **cout<<"Clubs";**

381 **}else if(card[i][0]==3){**

382 **cout<<"Spades";**

383 **}**

384 **}**

385 **}**

386 **int suit(int& randSuit){**

387 **randSuit=rand()%4+1;**

388 **return randSuit;**

389 **}**

390 \*/