**Project 1**

**<BlackJack>**

**CIS-5 48798**

**Name: Hunter Vint**

**Date: 11/5/2017**

**Intro:**

This program is a simplified version of blackjack, played against a computer. The player gets dealed two cards and has to get a value of cards adding up to or getting as close to 21 as possible. If the player goes over 21 he or she loses. The player closest to 21 gets the win for that round.

**Summary:**

Project size: about 250 lines

Number of variables: about 5

This project has many of the topics we examined in the first 8 weeks of class. It also has many areas in which it can be improved. I am not satisfied in the least with this project, and feel there are many areas I can improve in, but as my first major foray into coding I feel I did a decent job with trying to feel things out. Throughout this project I have learned many new strategies and methods through friends and research.

**Pseudocode:**

//System Libraries

//Input/Output Stream Library

//Standard Name-space under which System Libraries reside

//input output manipulator

//math library

//c standard library

//string library

//file input output library

//random number generator

//Function Prototypes

//Card Generation prototype

//output for the cards prototype

//check for win/loss conditions prototype

//opponent prototype

//Execution Begins Here!

//Variable Declaration (under 7 characters)

//set up each players hand

//dealer and player card declaration

//Variable Initialization

//explain rules

//cout for win for 21 on first hand or exact 21

//loss for any value over 21

//loss for value over 21 for computer or dealer

//choice of stay or hit for hand under 21

//recognition for capital and lowercase input of Hit

//Process Mapping - Only Inputs

//user input/data

//Output

//output into file "results"

//Exit function main/Exit the program

//prototypes begin here

//this prototype is for the card generator

//checks value of hands

//determines value of ace

//checks value of dealer hand

**Program:**

//System Libraries

#include <iostream> //Input/Output Stream Library

#include <iomanip>

#include <cmath>

#include <cstdlib>

#include <string>

#include <fstream>

#include <ctime>

using namespace std; //Standard Name-space under which System Libraries reside

//Function Prototypes

string crdGen(short); //Card Generation

string outCard(string); //output for the cards themselves (make a switch later)

short chekVal(string); //checks for win/loss conditions

short dealer(string); //opponent

//Execution Begins Here!

int main() {

//Variable Declaration

//keep these under 7 characters

ofstream out("Results.dat"); //output to file

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

string name, input = "hit";

string pCards = crdGen(2), dCards = crdGen(2); //sets up each players hand

short dTot = chekVal(dCards), pTot = chekVal(pCards); //dealer and player cards

//Variable Initialization

cout<<"Welcome to the BlackJack table!"<<endl; //explain rules

cout<<"The rules of blackjack are as follows:"<<endl;

cout<<"1) the goal of the game is to reach a collective value of 21."<<endl;

cout<<"2) the dealer will give you two cards to start with, and if you"

" want more you can say hit, otherwise say stay"<<endl;

cout<<"3) Number cards are worth face value, face cards are ten, and Aces"

" are either 1 or 11 (it works automatically)"<<endl;

cout<<"4) Be careful, for if the collective value of your cards"

" goes over 21, you lose."<<endl;

cout<<"5) if your opponent gets the same value you do, its a draw"<<endl;

cout<<"6) The person with the value closest to 21 will win!"<<endl;

cout<<endl;

cout<<"Shall we begin?"<<endl;

cout<<endl;

cout<<"What is your name?";

cin>>name;

cout<<endl;

cout<<"Hello, "<<name<<", today you will be playing"

" against our dealer!"<<endl;

do {

if (chekVal(pCards) == 21) { //21 on first hand or exact 21

cout<<"Congratulations! Your card value is 21 so you win!";

out<<"Congratulations! Your card value is 21 so you win!";

exit(EXIT\_SUCCESS); //works

}

else if (chekVal(pCards) > 21) { //any value over 21

cout<<"You Busted! You went over 21, so you lose!"<<endl;

out<<"You Busted! You went over 21, so you lose!"<<endl;

exit(EXIT\_SUCCESS); //works

}

else if (chekVal(dCards) > 21) { //same as above but for computer

cout<<"The Dealer busted! So you win!"<<endl;

out<<"The Dealer busted! So you win!"<<endl;

exit(EXIT\_SUCCESS); //this isn't working for some reason

}

else{ //hand under 21

cout<<"Your hand is "<<outCard(pCards)<<" which totals to "<<

chekVal(pCards)<<" would you like to hit "

"or stay?: "; //works and loops fine

out<<"Your hand is "<<outCard(pCards)<<" which totals to "<<

chekVal(pCards)<<" would you like to hit "

"or stay?: ";

cin>>input;

cout<<endl;

if (input == "hit" || input == "Hit") { //in case they use a capital

pCards += crdGen(1); //this took me a while to understand

}

else { //i didn't realize at first i only needed one condition here

dTot = dealer(dCards);

pTot = chekVal(pCards);

if (pTot > dTot) {

cout<<"Your total was "<<pTot<<" and the dealers"

" total was "<<dTot<<" So you win!";

out<<"Your total was "<<pTot<<" and the dealers"

" total was "<<dTot<<" So you win!";

exit (EXIT\_SUCCESS);

}

else if (pTot == dTot) {

cout<<"its a Draw! You and the dealer both had a total of "<<pTot;

out<<"its a Draw! You and the dealer both had a total of "<<pTot;

exit (EXIT\_SUCCESS);

}

else {

cout<<"You lost. Your total was "<<pTot<<" and the dealers"

" total was "<<dTot;

out<<"You lost. Your total was "<<pTot<<" and the dealers"

" total was "<<dTot;

exit (EXIT\_SUCCESS); //i think this one runs in place of

} //"dealer busted!"

}

}

}while (input == "Hit" || input == "hit");

//Process Mapping - Only Inputs

//user input/data

//Output

//Exit function main/Exit the program

out.close();

return 0;

}

//prototypes begin here...this was a headache and

//im not ashamed to admit i looked online for help

string crdGen (short numCrds) {

//this prototype is for the card generator

string temp = "";

for (short i = 0; i < numCrds; i++){

unsigned char temp2 = 0;

temp2 += (rand () % 13 + 1);

switch (temp2) {

case 1: temp += '2';break;

case 2: temp += '3';break;

case 3: temp += '4';break;

case 4: temp += '5';break;

case 5: temp += '6';break;

case 6: temp += '7';break;

case 7: temp += '8';break;

case 8: temp += '9';break;

case 9: temp += '0';break;

//0 stands for 10, because char

case 10: temp += 'J';break;

case 11: temp += 'Q';break;

case 12: temp += 'K';break;

case 13: temp += 'A';break;

}

}

return temp;

}

short chekVal (string hand) {

//checks value of hands

short total = 0;

short hAce = 0;

for (short i = 0; i < hand.length (); i++) {

char temp = hand [i];

switch (temp) {

case '2': total += 2;break;

case '3': total += 3;break;

case '4': total += 4;break;

case '5': total += 5;break;

case '6': total += 6;break;

case '7': total += 7;break;

case '8': total += 8;break;

case '9': total += 9;break;

case '0': total += 10;break;

case 'J': total += 10;break;

case 'Q': total += 10;break;

case 'K': total += 10;break;

case 'A': hAce += 1;break;

}

}

for (short i = 0; i < hAce; i++) {

//determines value of an Ace

((total + 11)> 21? (total += 1) : (total += 11));

}

return total;

}

string outCard (string hand) {

string outPut = "";

for (short i = 0; i < hand.length (); i++) {

char temp = hand [i];

switch (temp){

//sorry about going over the line a bit here

case '2': (i == hand.length () - 1)?(outPut += "2"):(outPut += "2, ")

;break;

//needed to make sure it doesn't initialize them all

case '3': (i == hand.length () - 1)?(outPut += "3"):(outPut += "3, ")

;break;

case '4': (i == hand.length () - 1)?(outPut += "4"):(outPut += "4, ")

;break;

case '5': (i == hand.length () - 1)?(outPut += "5"):(outPut += "5, ")

;break;

case '6': (i == hand.length () - 1)?(outPut += "6"):(outPut += "6, ")

;break;

case '7': (i == hand.length () - 1)?(outPut += "7"):(outPut += "7, ")

;break;

case '8': (i == hand.length () - 1)?(outPut += "8"):(outPut += "8, ")

;break;

case '9': (i == hand.length () - 1)?(outPut += "9"):(outPut += "9, ")

;break;

case '0': (i == hand.length () - 1)?(outPut += "10"):(outPut += "10, ")

;break;

case 'J': (i == hand.length () - 1)?(outPut += "J"):(outPut += "J, ")

;break;

case 'Q': (i == hand.length () - 1)?(outPut += "Q"):(outPut += "Q, ")

;break;

case 'K': (i == hand.length () - 1)?(outPut += "K"):(outPut += "K, ")

;break;

case 'A': (i == hand.length () - 1)?(outPut += "A"):(outPut += "A, ")

;break;

}

}

return outPut;

}

short dealer (string hand) {

short val = chekVal(hand);

if (val > 21) {

return 22;

}

else if (val < 17) {

do {

val = chekVal(hand);

hand += crdGen(1);

}while(val < 17);

}

return val;

}

|  |  |  |  |
| --- | --- | --- | --- |
| **Cross Reference for Project 1** | | | |
|  |  |  |  |
|  |  |  | **Where in Code** |
| **Chapter** | **Section** | **Topic** | **Line number** |
| 2 | 2 | cout | 52 |
|  | 3 | libraries | iostream, iomanip, cmath, cstdlib, fstream, string, ctime |
|  | 4 | variables/literals | 44 |
|  | 5 | Identifiers | 148 |
|  | 6 | Integers | 148 |
|  | 7 | Characters | 174 |
|  | 8 | Strings | 169 |
|  | 9 | Floats No Doubles |  |
|  | 10 | Bools |  |
|  | 11 | Sizeof \*\*\*\*\* |  |
|  | 12 | Variables 7 characters or less | 44 |
|  | 13 | Scope \*\*\*\*\* No Global Variables | 87 |
|  | 14 | Arithmetic operators | 148 |
|  | 15 | Comments 20%+ | 141 |
|  | 16 | Named Constants |  |
|  | 17 | Programming Style \*\*\*\*\* Emulate |  |
|  |  |  |  |
| 3 | 1 | cin | 67 |
|  | 2 | Math Expression | 171 |
|  | 3 | Mixing data types \*\*\*\* |  |
|  | 4 | Overflow/Underflow \*\*\*\* |  |
|  | 5 | Type Casting |  |
|  | 6 | Multiple assignment \*\*\*\*\* |  |
|  | 7 | Formatting output | 233 |
|  | 8 | Strings | 143 |
|  | 9 | Math Library | 148 |
|  | 10 | Hand tracing \*\*\*\*\*\* |  |
|  |  |  |  |
| 4 | 1 | Relational Operators |  |
|  | 2 | if | 96 |
|  | 4 | If-else | 96 |
|  | 5 | Nesting |  |
|  | 6 | If-else-if | 102 |
|  | 7 | Flags \*\*\*\*\* |  |
|  | 8 | Logical operators | 96 |
|  | 11 | Validating user input | 123 |
|  | 13 | Conditional Operator | 109 |
|  | 14 | Switch | 149 |
|  |  |  |  |
| 5 | 1 | Increment/Decrement |  |
|  | 2 | While | 123 |
|  | 5 | Do-while | 71 |
|  | 6 | For loop | 146 |
|  | 11 | Files input/output both | 74 |
|  | 12 | No breaks in loops \*\*\*\*\*\* | 71 |
|  |  |  |  |
|  |  |  |  |
| \*\*\*\*\*\* Not required to show |  |  |  |