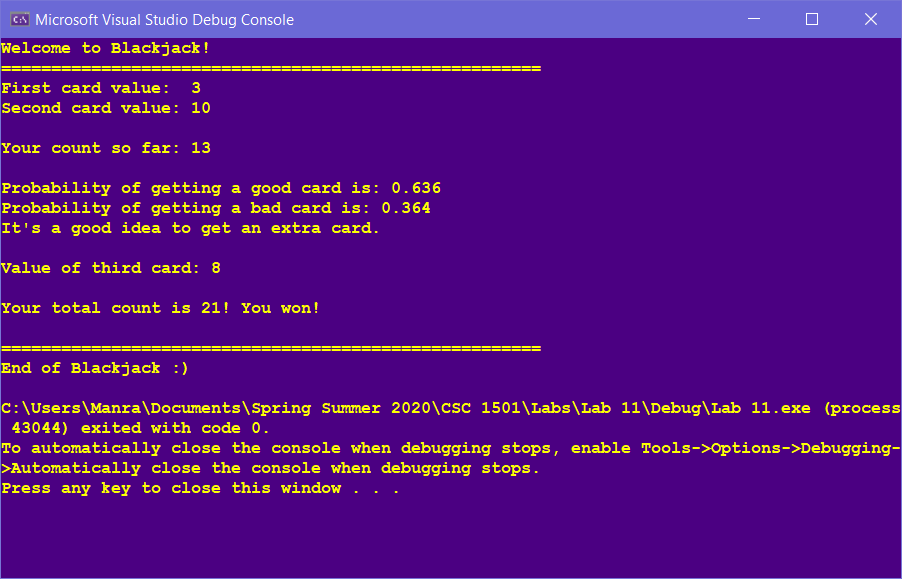
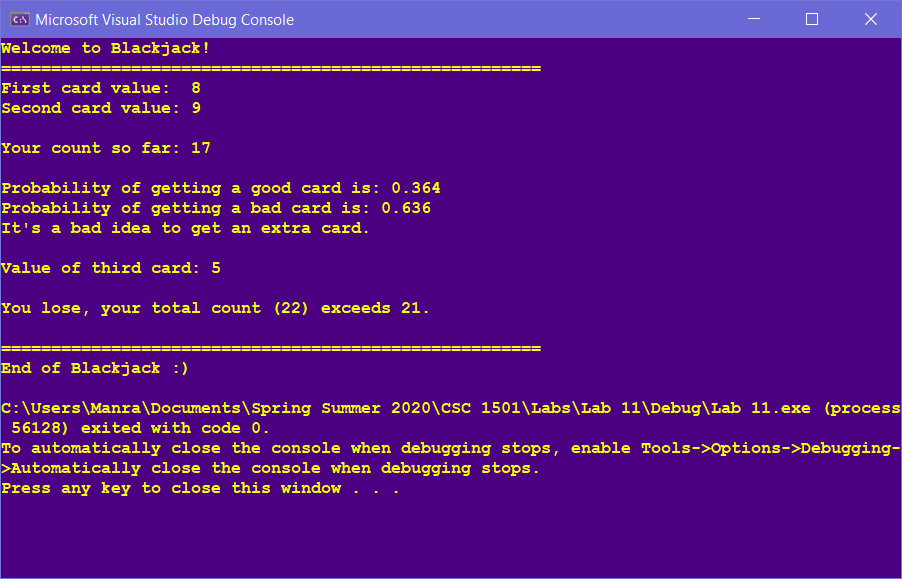
Lab 11





**//==========================================================**

**//**

**// Title: Blackjack**

**// Course: CSC 1501**

**// Lab Number: 11**

**// Author: Manraj Singh**

**// Date: 07/13/2020**

**// Description:**

**// This lab creates a Blackjack game.**

**//**

**//==========================================================**

**#include <cstdlib> // For several general-purpose functions**

**#include <fstream> // For file handling**

**#include <iomanip> // For formatted output**

**#include <iostream> // For cin, cout, and system**

**#include <string> // For string data type**

**using namespace std; // So "std::cout" may be abbreviated to "cout"**

**int main(){**

**//declare variables**

**int card1; //1st card**

**int card2; //2nd card**

**int card3; //3rd card**

**int count; //sum of card values**

**int ace1 = 1; //user choice if they want ace to be 1**

**int ace11 = 11; //user choice if they want ace to be 11**

**char response; //user response to choose ace value**

**double goodProb = 1.0; //good probability variable**

**double badProb = 0.0; //bad probability variable**

**//random number seeder**

**srand(time(NULL));**

**//probability decimal placed to 3**

**cout << fixed << setprecision(3);**

**//start of application**

**cout << "Welcome to Blackjack!" << endl;**

**cout << "======================================================" << endl;**

**cout << "First card value: ";**

**//gives card 1 a random value from 1-11**

**card1 = (rand() % 11) + 1;**

**//if a card is 1 or 11, it is an ace, and the user chooses which one they want**

**if (card1 == ace1 || card1 == ace11) {**

**cout << "You received an Ace card. \nEnter 'A' if you want the value to be 1, or 'B' if you want the value to be 11." << endl;**

**cin >> response;**

**if (response == 'A')**

**card1 = 1;**

**else if (response == 'B')**

**card1 = 11;**

**cout << card1 << endl;**

**}**

**else**

**cout << card1 << endl;**

**cout << "Second card value: ";**

**//gives card 2 a random value from 1-11**

**card2 = (rand() % 11) + 1;**

**//while card 2 is equal to card 1 it is randomized until it is different. If it is 10 it is outputted (4 cards = 10)**

**while (card2 == card1) {**

**if (card2 == 10) {**

**cout << card2 << endl << endl;**

**}**

**else {**

**card2 = (rand() % 11) + 1;**

**//checks with user for their ace choice**

**if (card1 == ace1 || card1 == ace11) {**

**if (card2 == ace1 || card2 == ace11) {**

**card2 = (rand() % 11) + 1;**

**cout << card2 << endl << endl;**

**}**

**}**

**else**

**cout << card2 << endl << endl;**

**}**

**}**

**//if card 1 wasn't an ace and card 2 is, it gives the user a choice to choose 1 or 11**

**if (card2 == ace1 || card2 == ace11) {**

**cout << "You received an Ace card. \nEnter 'A' if you want the value to be 1, or 'B' if you want the value to be 11.";**

**cin >> response;**

**if (response == 'A') {**

**card2 = 1;**

**cout << card2 << endl << endl;**

**}**

**else if (response == 'B') {**

**card2 = 11;**

**cout << card2 << endl << endl;**

**}**

**}**

**else**

**cout << card2 << endl << endl;**

**//sum is explicitly casted to double for probability calculation**

**count = (double)card1 + card2;**

**cout << "Your count so far: " << count << endl << endl;**

**//if 2 cards give 21, automatically win**

**if (count == 21) {**

**cout << "YOU WON! You got an ace and a 10! Congrats!" << endl;**

**//end of application**

**cout << "\n======================================================" << endl;**

**cout << "End of Blackjack :)" << endl;**

**return 0;**

**}**

**//if the sum of 2 cards is 11 or less, then any card given will not lose the game, so the probability of a good card is 100%**

**if (count <= 11) {**

**//output probability and tell user it is a good idea to get a 3rd card**

**goodProb = 1.0;**

**badProb = 0.0;**

**cout << "\nProbability of getting a good card is: " << goodProb \* 100 << "%" << endl;**

**cout << "Probability of getting a bad card is: " << badProb \* 100 << "%" << endl;**

**cout << "It's a good idea to take a third card because you won't lose in any case." << endl << endl;**

**//randomize third card and output**

**cout << "Value of third card: " << endl;**

**card3 = (rand() % 11) + 1;**

**//check if 3rd card is equal to 1st or 2nd, and if not 10, is re-randomized and outputted**

**while (card3 == card1 || card3 == card2) {**

**if (card3 == 10) {**

**cout << card3 << endl;**

**}**

**else {**

**card3 = (rand() % 11) + 1;**

**cout << card3 << endl;**

**}**

**}**

**//checks with user for ace value choice**

**if (card3 == ace1 || card3 == ace11) {**

**cout << "You received an Ace card. \nEnter 'A' if you want the value to be 1, or 'B' if you want the value to be 11.";**

**cin >> response;**

**if (response == 'A') {**

**card3 = 1;**

**cout << card3 << endl;**

**}**

**else if (response == 'B') {**

**card3 = 11;**

**cout << card3 << endl;**

**}**

**}**

**else**

**cout << card3 << endl;**

**//sum updated**

**count = count + card3;**

**//if sum is less than 21, they are still in the game**

**if (count < 21)**

**cout << "Your count is less than 21 (" << count << ") and you are still in the game" << endl;**

**//if sum is 21, they win**

**else if (count == 21)**

**cout << "You won!" << endl;**

**}**

**//if sum is greater than 11, then the odds change, and I use a switch case to make it easier to see each possible case**

**else {**

**if (count == 12) {**

**//if sum is 12, calculate probability and output their chances**

**//if both cards are less than the needed value, exclude them from calculating good probability**

**if (card1 <= (21 - count) && card2 <= (21 - count)) {**

**goodProb = ((21. - count) - 2) / 11;**

**badProb = (11. - (goodProb \* 11)) / 11;**

**}**

**//if one card is part less than needed value, and one is more than needed value, exclude both from respective probabilities**

**else if (card1 <= (21 - count) && card2 > (21 - count)) {**

**goodProb = ((21. - count) - 1) / 11;**

**badProb = (11. - (goodProb \* 11)) / 11;**

**}**

**//same as above, but checks for opposite cards**

**else if (card1 > (21 - count) && card2 < (21 - count)) {**

**goodProb = ((21. - count) - 1) / 11;**

**badProb = (11. - (goodProb \* 11)) / 11;**

**}**

**cout << "Probability of getting a good card is: " << goodProb << endl;**

**cout << "Probability of getting a bad card is: " << badProb << endl;**

**//if the probability is greater than 0.5, then it is a good idea to go for a 3rd card.**

**if (goodProb > 0.5)**

**cout << "It's a good idea to get an extra card." << endl << endl;**

**else**

**cout << "It's a bad idea to get an extra card." << endl << endl;**

**}**

**if (count > 12 && count <= 20) {**

**//if both cards are less than the needed value, exclude them from calculating good probability**

**if (card1 <= (21 - count) && card2 <= (21 - count)) {**

**goodProb = ((21. - count) - 2) / 11;**

**badProb = (11. - (goodProb \* 11)) / 11;**

**}**

**//if one card is part less than needed value, and one is more than needed value, exclude both from respective probabilities**

**else if (card1 <= (21 - count) && card2 > (21 - count)) {**

**if (card2 == 11) {**

**goodProb = ((21. - count) - 2) / 11;**

**badProb = (11. - (goodProb \* 11)) / 11;**

**}**

**goodProb = ((21. - count) - 1) / 11;**

**badProb = (11. - (goodProb \* 11)) / 11;**

**}**

**//same as above, but checks for opposite cards**

**else if (card1 > (21 - count) && card2 <= (21 - count)) {**

**if (card1 == 11) {**

**goodProb = ((21. - count) - 2) / 11;**

**badProb = (11. - (goodProb \* 11)) / 11;**

**}**

**goodProb = ((21. - count) - 1) / 11;**

**badProb = (11. - (goodProb \* 11)) / 11;**

**}**

**//if both cards are greater than the difference, then this is outputted**

**else if (card1 > (21 - count) && card2 > (21 - count)) {**

**//if either card is 11, 1 is removed from the goodProb because ace is go**

**if (card1 == 11 || card2 == 11) {**

**goodProb = ((21. - count) - 1) / 11;**

**badProb = (11. - (goodProb \* 11)) / 11;**

**}**

**goodProb = ((21. - count)) / 11;**

**badProb = (11. - (goodProb \* 11)) / 11;**

**}**

**cout << "Probability of getting a good card is: " << goodProb << endl;**

**cout << "Probability of getting a bad card is: " << badProb << endl;**

**if (goodProb > 0.5)**

**cout << "It's a good idea to get an extra card." << endl << endl;**

**else**

**cout << "It's a bad idea to get an extra card." << endl << endl;**

**}**

**//gives random value for 3rd card**

**cout << "Value of third card: ";**

**card3 = (rand() % 11) + 1;**

**//check if 3rd card is equal to 1st or 2nd, and if not 10, is re-randomized and outputted**

**while (card3 == card1 || card3 == card2) {**

**if (card3 == 10) {**

**cout << card3 << endl << endl;**

**}**

**else {**

**card3 = (rand() % 11) + 1;**

**//checks with user for their ace choice**

**if (card1 == ace1 || card1 == ace11 || card2 == ace1 || card2 == ace11) {**

**if (card3 == ace1 || card3 == ace11) {**

**card3 = (rand() % 11) + 1;**

**}**

**}**

**}**

**}**

**//if card 3 is an ace and 1 or 2 weren't, then the user is given the ace choice**

**if (card3 == ace1 || card3 == ace11) {**

**cout << "You received an Ace card. \nEnter 'A' if you want the value to be 1, or 'B' if you want the value to be 11.";**

**cin >> response;**

**if (response == 'A') {**

**card3 = 1;**

**cout << card3 << endl << endl;**

**}**

**else if (response == 'B') {**

**card3 = 1;**

**cout << card3 << endl << endl;**

**}**

**}**

**else**

**cout << card3 << endl << endl;**

**count = count + card3;**

**//if sum is < 21, they can still play**

**if (count < 21)**

**cout << "Your count is less than 21 (" << count << ") so you are still in the game" << endl;**

**else if (count == 21)**

**cout << "Your total count is 21! You won!" << endl;**

**else {**

**//tells user they lose if the sum goes over 21**

**cout << "You lose, your total count (" << count << ") exceeds 21." << endl;**

**}**

**}**

**//end of application**

**cout << "\n======================================================" << endl;**

**cout << "End of Blackjack :)" << endl;**

**return 0;**

**}**