Project 1

Jamil Batshoun

CIS – 5

**Introduction**

Title: Baccarat

Baccarat can be played with a minimum of two people. At a casino, the players would be referred to as the house and the player. In Baccarat, each player is given two cards, which are worth points. Whoever has the most points wins. How many points a player has is determined by the sum of their cards. However, if the cards add up to ten or greater, then the points are worth the value of the right most digit of the sum of the cards. For instance, cards with a face value of 9 and 7 have a sum of 16, however, the player will only have a score of 6. Therefore, the highest possible winning sum would be a score of 9. Cards two through nine are worth their face value in points, while a king, queen, or jack are worth zero points. Aces are worth one point.

**Summary**

Project size: 148 lines

Number of variables: 20

In this program a file is open which holds the numbers zero through nine to represent the possible points a card is worth. Next a random number is generated which when added to a for loop will decide how many lines are read from that file. The last line read from that file will be a number held in the “hold” type of variables that is passed to a function where it is switched for a different number. That new number will represent the points on one of four cards. The process is repeated three more time to generate a total of four “cards.” The first two are added and assigned to the user. The second two are assigned to the “house.” If the cards add up to ten or more, then ten is subtracted from the sum. The sums are compared to see which is highest and who’s sum it was. If the player wins, a counter is incremented. The player is then asked if they want to play again. If the user wants to play again, a separate counter is incremented and then a “chance” of winning the next match is calculated. Over time this “chance” will approach 50%, given as the outcome of the game is random.

**Description**

The main ideas of this program are working with random numbers and the handling and reading in of a file.

**Flowchart Link**

https://www.gliffy.com/go/publish/11772404

|  |  |  |  |
| --- | --- | --- | --- |
| **Cross Reference for Project 1** | | | |
|  |  |  | **Where in Code** |
| **Chapter** | **Section** | **Topic** | **Line number** |
| 2 | 2 | cout | 36 |
|  | 3 | libraries | 8 ->13 |
|  | 4 | variables/literals | 21 |
|  | 5 | Identifiers | 21 |
|  | 6 | Integers | 23 |
|  | 7 | Characters | 21 |
|  | 8 | Strings | 32 |
|  | 9 | Floats No Doubles | 27 |
|  | 10 | Bools | 38 |
|  | 11 | Sizeof \*\*\*\*\* |  |
|  | 12 | Variables 7 characters or less | 21 |
|  | 13 | Scope \*\*\*\*\* No Global Variables |  |
|  | 14 | Arithmetic operators | 40 |
|  | 15 | Comments 20%+ | 22 |
|  | 16 | Named Constants | 30 |
|  | 17 | Programming Style \*\*\*\*\* Emulate |  |
|  |  |  |  |
| 3 | 1 | cin | 125 |
|  | 2 | Math Expression | 40 |
|  | 3 | Mixing data types \*\*\*\* |  |
|  | 4 | Overflow/Underflow \*\*\*\* |  |
|  | 5 | Type Casting |  |
|  | 6 | Multiple assignment \*\*\*\*\* |  |
|  | 7 | Formatting output | 36,93 ->93 |
|  | 8 | Strings | 31 |
|  | 9 | Math Library |  |
|  | 10 | Hand tracing \*\*\*\*\*\* |  |
|  |  |  |  |
| 4 | 1 | Relational Operators | 50 |
|  | 2 | if | 88 |
|  | 4 | If-else | 103 ->108 |
|  | 5 | Nesting | 103 |
|  | 6 | If-else-if | 102 ->111 |
|  | 7 | Flags \*\*\*\*\* |  |
|  | 8 | Logical operators | 111 |
|  | 11 | Validating user input |  |
|  | 13 | Conditional Operator | 100 |
|  | 14 | Switch | 134 |
|  |  |  |  |
| 5 | 1 | Increment/Decrement | 50 |
|  | 2 | While |  |
|  | 5 | Do-while | 36 |
|  | 6 | For loop | 50 |
|  | 11 | Files input/output both | 32 ->34 |
|  | 12 | No breaks in loops \*\*\*\*\*\* |  |
| \*\*\*\*\*\* Not required to show |  |  |  |

**Pseudocode**

//system libraries: iostream cstdlib fstream string ctime and iomanip

//namespace

//declare the switchr function with one interger argument

//main function

//declare char variable “again”

//declare int variables for1, for2, for3, and for4

//declare int variables hold1, hold2, hold3, hold4, yrScore, and hsScore

//declare float nWins, avg, and nGames. Initialize nGames to 0

//declare constant integers min initialized to 1 and max initialized to 10

//declare string fileNme and initialize to “Cards.dat”

//declare ifstream variable inpFile

//open input file “Cards.dat”

//set random number seed to time(0)

//print game name to screen

//begin do-while loop

//declare bool variable didWin

//set for1 to rand( ) %(max minus min plus 1) plus min

//set for2 to rand( ) %(max minus min plus 1) plus min

//set for3 to rand( ) %(max minus min plus 1) plus min

//set for4 to rand( ) %(max minus min plus 1) plus min

//open input file

//read from input file for1 number of times. Last line is the value of hold1

//hold1 is the return value of switchr function with an argument of hold1

//close file

//open file

//read from input file for2 number of times. Last line is the value of hold2

//hold2 is the return value of switchr function with an argument of hold2

//close file

//open file

//read from input file for3 number of times. Last line is the value of hold3

//hold3 is the return value of switchr function with an argument of hold3

//close file

//open file

//read from input file for4 number of times. Last line is the value of hold4

//hold4 is the return value of switchr function with an argument of hold4

//close file

//set yrScore equal to the sum of hold1 and hold2

//set hsScore equal to the sum of hold3 and hold4

//if yrScore is greater than 10

//subtract 10 from yrScore

//if hsScore is greater than 10

//subtract 10 from hsScore

//print out “Your cards” and “The House’s cards”

//print out hold1, hold2, hold3, and hold4

//print out “Your score” and “The House’s score”

//print out yrScore and hsScore

// if yrScore is greater than or equal to hsScore, set didWin to true, else false

//if didWin is true

//if yrScore is not equal to hsScore

//print out “You Won!”

//increment nWins

//else

//print out “Tie”

//else if yrScore is less than hsScore

//print out “You lost”

//increment nGames

//set avg to (nWins divided by nGames) and multiply by 100

//use fixed, setprecision(2), and showpoint

//print out “Odds of winning = ,” avg, and then “%”

//print out “Play again? Press y to play again. Press any other key to exit:”

//let user enter in the value of char variable “again”

//end do while loop if “again” is equal to y or Y

//print out “Thanks for playing”

//return 0

//declare switchr function with a integer variable of holder

//switch holder

//if holder is 9 then make holder 0

//if holder is 8 then make holder 1

//if holder is 7 then make holder 2

//if holder is 6 then make holder 3

//if holder is 5 then make holder 4

//if holder is 4 then make holder 5

//if holder is 3 then make holder 6

//if holder is 2 then make holder 7

//if holder is 1 then make holder 8

//if holder is 0 then make holder 9

//return holder