Project 1

CIS-5

41595

Name: Avila, Giselle

Date: 2/6/2023

Introduction

Title: Black Jack

A deck of cards is shuffled. Each player gets two cards while the dealer's second card is face down. The players can decide to hit or stay depending on how they feel about their cards. If you hit, a new card is given to you and that value will add to your overall total. If you stay, it is the dealer's turn to hit cards. If the player goes over 21, they automatically lose. If the dealer goes over 21, the player automatically wins. The goal is to get the highest value or 21 and not go over 21. The highest value without exceeding 21 wins.

Summary

Project size: 320 lines

Number of variables: about 14

I completed the project over the past three days. It was a time crunch, but I managed to input a good majority of my learning within that short time. I had many troubles where I didn't include splitting and doubling down in the game and even the choice of the ace card. Due to a heavy schedule, I was not able to translate it into code. I tried my best to reflect my learning into this game.

Description

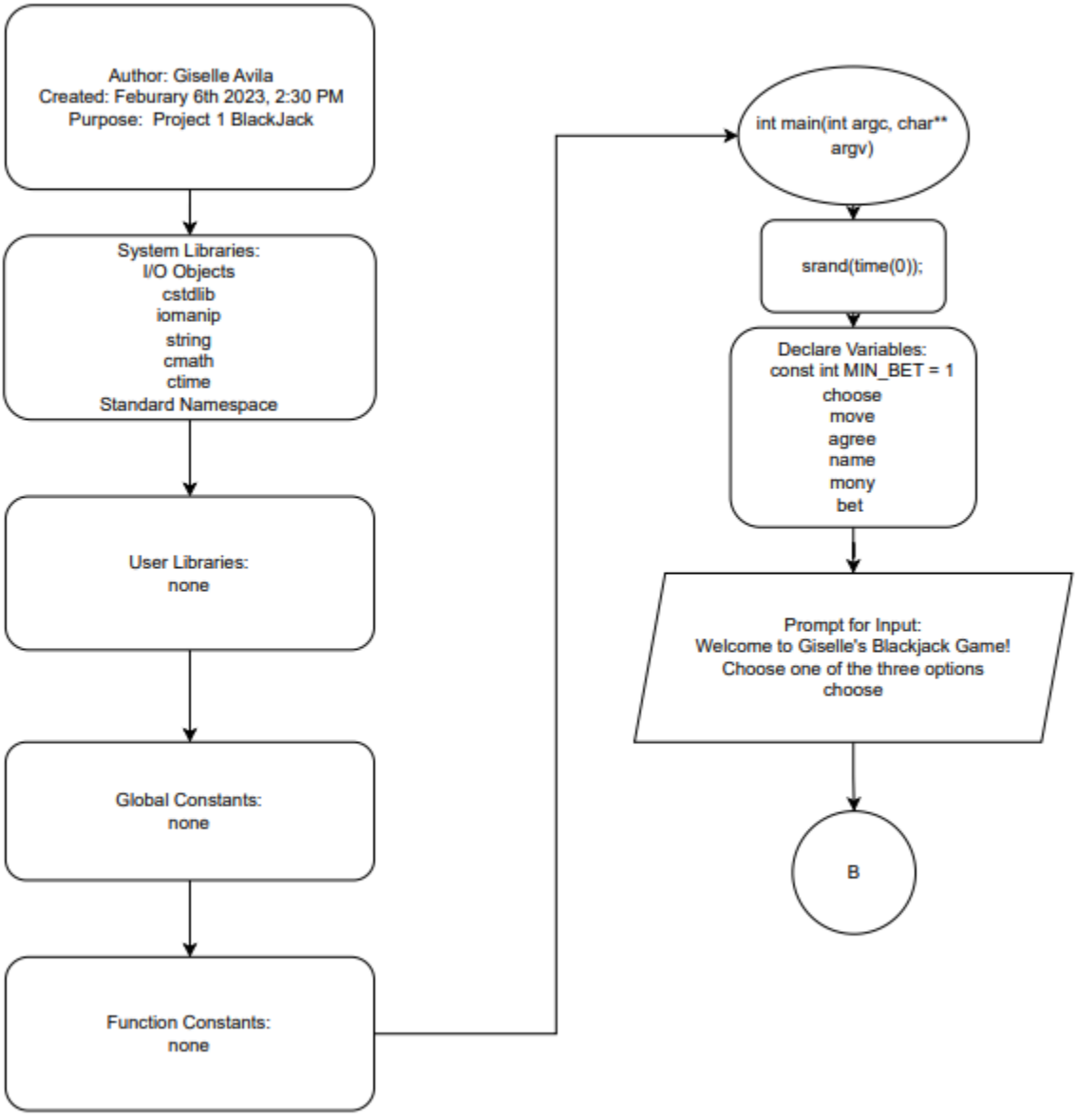
Each hand begins with giving everyone two cards. All cards are dealt face up except for the dealer's last card. Players act first with two options: Hitting(Tap), Standing(Wave). Hitting: You want another card so you tap the table. Players can tap as many times as they want until they stand(are happy with their hand) or when they go bust. Standing: You wave to show that you are happy with your hand, therefore will not have another card. Once the player is done with hitting, it is the dealer’s choice to hit or not now. If the dealer and the player have the same value then it is a tie. There is no winner. If the dealer or player exceeds 21, they lose automatically. If the player or dealer gets 21, they win. If no one gets 21, then the player or dealer with the highest value wins.

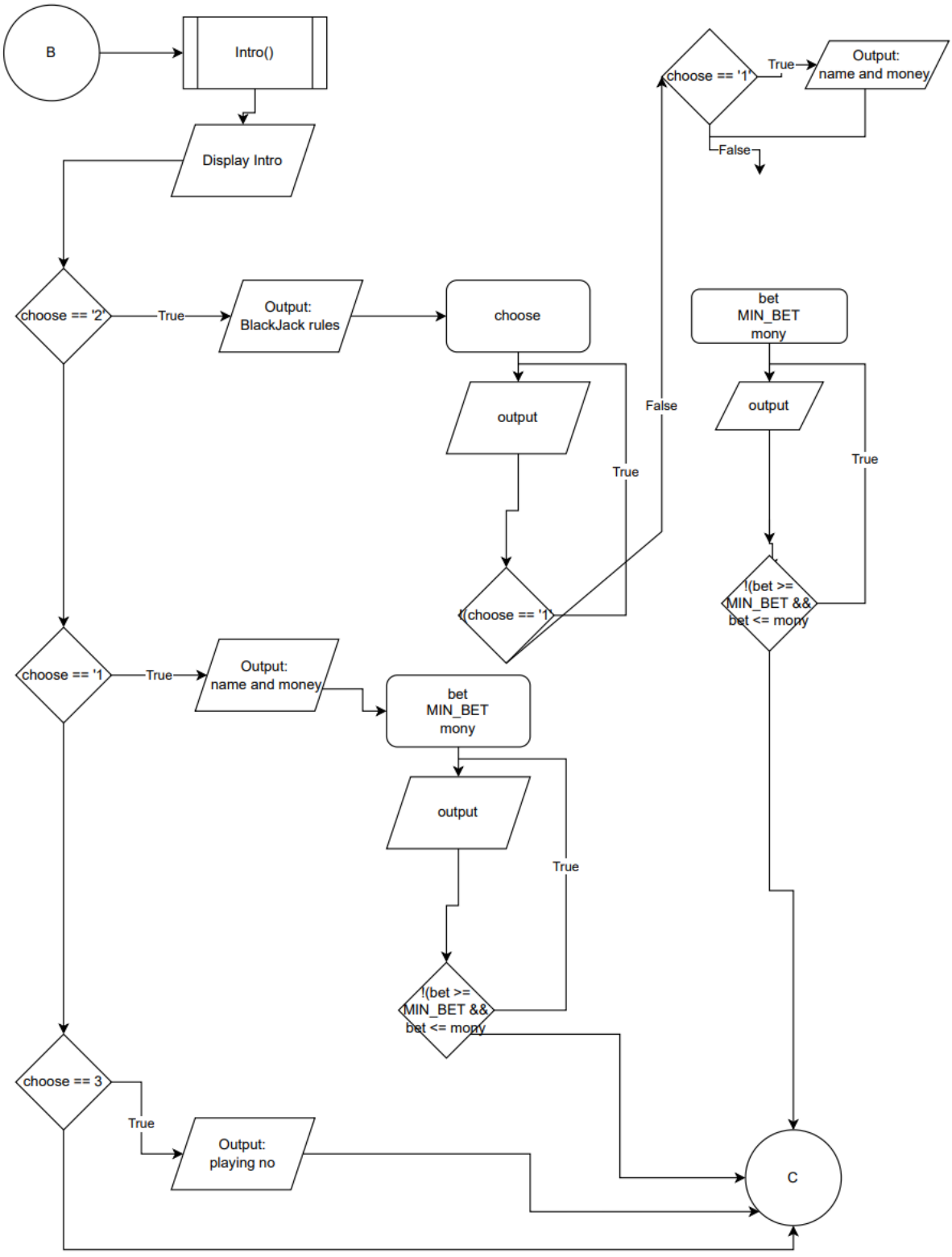
Check List

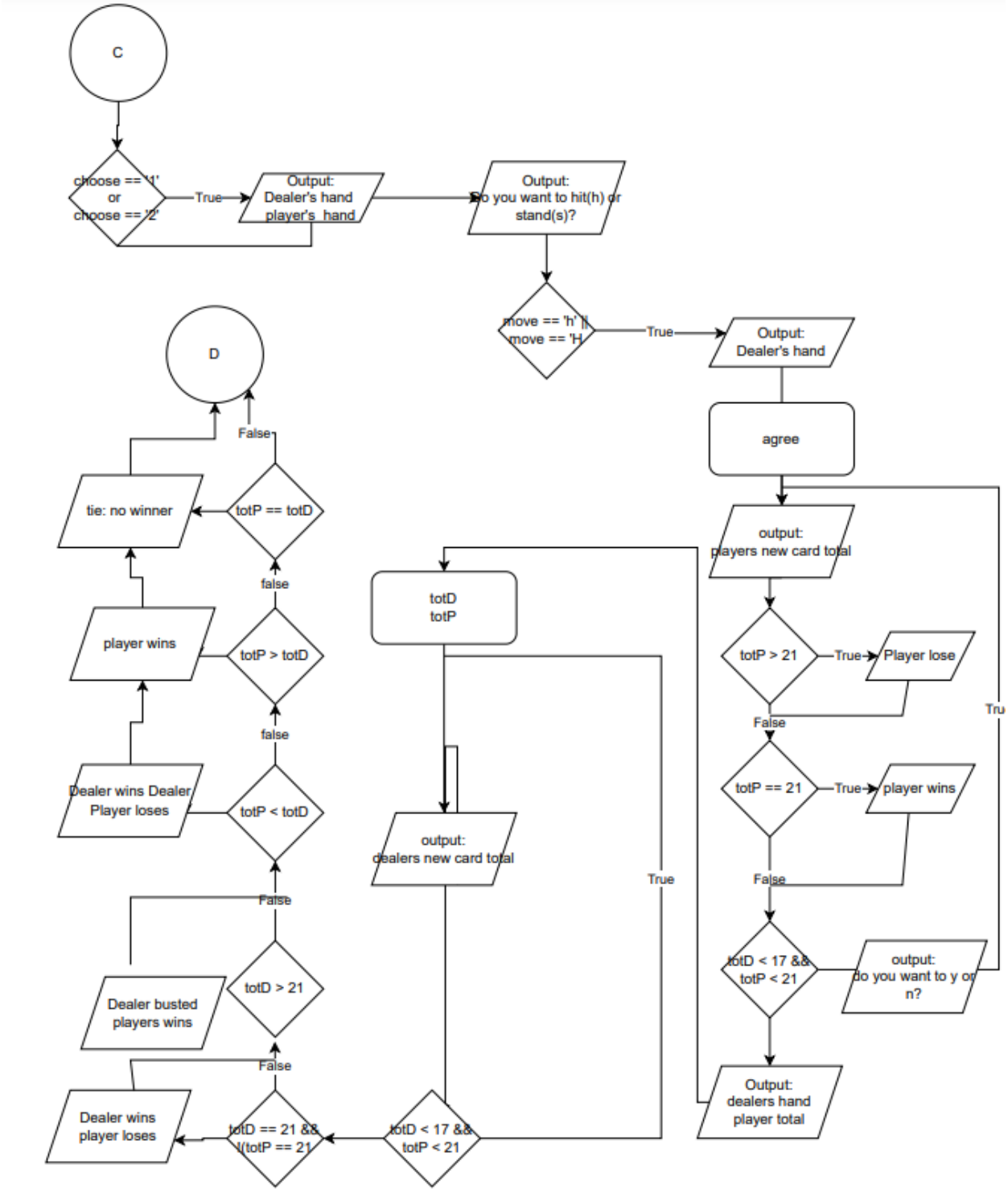
Chapter	Section	Topic	Where Line #"s	Pts	Notes	
2	2	cout				X
	3	libraries		8	iostream, iomanip, cmath, cstdlib, fstream, string, ctime	X
	4	variables/literals			No variables in global area, failed project!	
	5	Identifiers				
	6	Integers		3		X
	7	Characters		3		X
	8	Strings		3		X
	9	Floats No Doubles		3	Using doubles will fail the project, floats OK!	X
	10	Bools		4		
	11	Sizeof *****				
	12	Variables 7 characters or less			All variables <= 7 characters	X
	13	Scope ***** No Global Variables				
	14	Arithmetic operators				
	15	Comments 20%+		5	Model as pseudo code	X
	16	Named Constants			All Local, only Conversions/Physics/Math in Global area	
	17	Programming Style ***** Emulate			Emulate style in book/in class repositiory	
3	1	cin				
	2	Math Expression				
	3	Mixing data types ****				
	4	Overflow/Underflow ****				
	5	Type Casting		4		
	6	Multiple assignment *****				
	7	Formatting output		4		X
	8	Strings		3		X
	9	Math Library		4	All libraries included have to be used	X

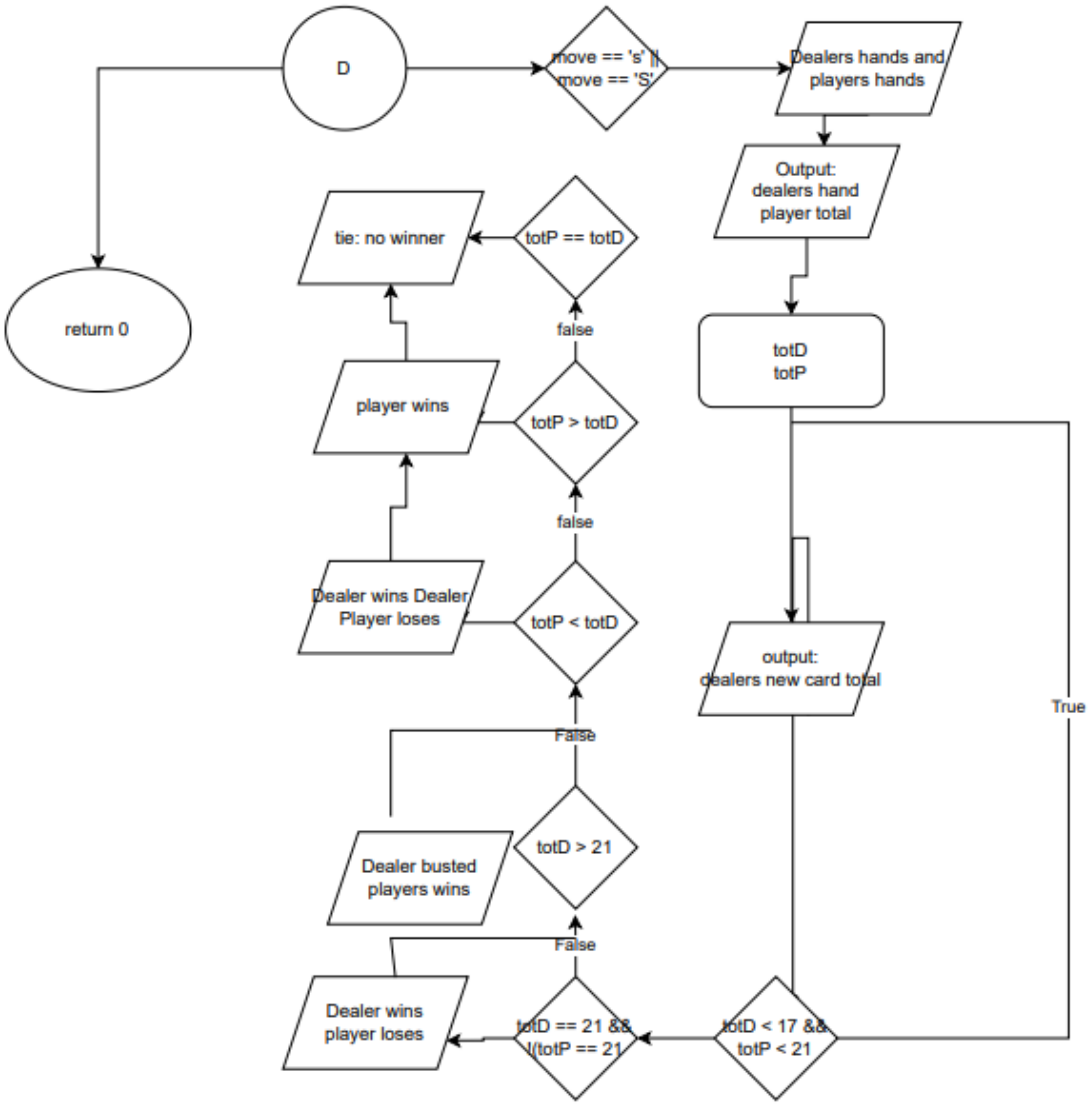
4	1	Relational Operators				
	2	if		4	Independent if	X
	4	If-else		4		X
	5	Nesting		4		X
	6	If-else-if		4		
	7	Flags *****				
	8	Logical operators		4		X
	11	Validating user input		4		
	13	Conditional Operator		4		X
	14	Switch		4		X
5	1	Increment/Decrement		4		
	2	While		4		X
	5	Do-while		4		X
	6	For loop		4		
	11	Files input/output both		8		
	12	No breaks in loops ******			Failed Project if included	
****** Not required to show			Total	100		

Flowchart


Author: Giselle Avila
Created: Feburary 6th 2023, 2:30 PM
Purpose: Project 1 BlackJack
System Libraries:
I/O Objects
cstdlib
iomanip
string
cmath
ctime
Standard Namespace
User Libraries:
none
Global Constants:
none
Function Constants:
none
int main(int argc, char** argv)
srand(time(0));
Declare Variables:
const int MIN_BET = 1
choose
move
agree
name
mony
bet
Prompt for Input:
Welcome to Giselle's Blackjack Game!
Choose one of the three options
choose
B


B
Intro()
Display Intro
choose == '2'
True
Output:
BlackJack rules
choose
output
!(choose == '1'
False
True
choose == '1'
True
Output:
name and money
False
bet
MIN_BET
mony
output
!(bet >=
MIN_BET &&
bet <= mony
True
choose == '1
True
Output:
name and money
bet
MIN_BET
mony
output
!(bet >=
MIN_BET &&
bet <= mony
True
choose == 3
True
Output:
playing no
C


C
choose == '1'
or
choose == '2'
True
Output:
Dealer's hand
player's hand
Output:
Do you want to hit(h) or
stand(s)?
move == 'h' ||
move == 'H'
True
Output:
Dealer's hand
agree
output:
players new card total
totP > 21
True
Player lose
False
totP == 21
True
player wins
False
totD < 17 &&
totP < 21
output:
do you want to y or
n?
Tru
Output:
dealers hand
player total
totD
totP
output:
dealers new card total
True
totD < 17 &&
totP < 21
totD == 21 &&
!(totP == 21
Dealer wins
player loses
False
totD > 21
Dealer busted
players wins
False
totP < totD
Dealer wins Dealer
Player loses
false
totP > totD
player wins
false
totP == totD
tie: no winner
False
D


D
move == 's' ||
move == 'S'
Dealers hands and
players hands
Output:
dealers hand
player total
totD
totP
output:
dealers new card total
totD < 17 &&
totP < 21
True
totD == 21 &&
!(totP == 21
Dealer wins
player loses
False
totD > 21
Dealer busted
players wins
False
totP < totD
Dealer wins Dealer
Player loses
false
totP > totD
player wins
false
totP == totD
tie: no winner
return 0

Pseudo Code

Initialize

Menu of the 3 options prints out, player chooses one of the choices

If player chooses 1

Print out the question for the player to insert name & money amount

Get bet amount from player based on money amount and min bet

If player chooses 2

Print rules of Blackjack

Print out option to begin game by typing 1

If player types 1 to begin game

Print out the question for the player to insert name & money amount

Get bet amount from player based on money amount & min bet

If player chooses 3

Print Goodbye, Game will not continue

If player chooses one of the choices from menu 1 or 2

Game play begins

Dealer's hand is printed except for the second card

Player's hand is printed

Player's move to hit or stay

If player chooses to hit

Print Dealer's second card

New card is printed and added to player's total

If player's total is greater than 21

Print Bust! Dealer wins!

If player's total is equal to 21

Print Winner! Dealer loses!

Game exited

Player choice to hit again

If yes repeat process again

If no player stays

Dealer's turn to hit or stay

Dealer hits when total under 17

If Dealer total equal to 21 and player's total is not 21

Print Dealer wins! Player loses!

If Dealer total is over 21

Print Dealer busted! Player Wins!

Else if players total less than dealer's total

Print player lost! Dealer wins!

Else if players total greater than dealer's total

Print player wins! Dealer loses!

Else if players total is the same as dealers total

Print it's a tie! No winner!

Game ends

If player chooses to stay

Print Dealer's second card

Dealer's turn to hit or stay

Dealer hits when total under 17

If Dealer total equal to 21 and player's total is not 21

Print Dealer wins! Player loses!

If Dealer total is over 21

Print Dealer busted! Player Wins!

Else if players total less than dealer's total

Print player lost! Dealer wins!

Else if players total greater than dealer's total

Print player wins! Dealer loses!

Else if players total is the same as dealers total

Print it's a tie! No winner!

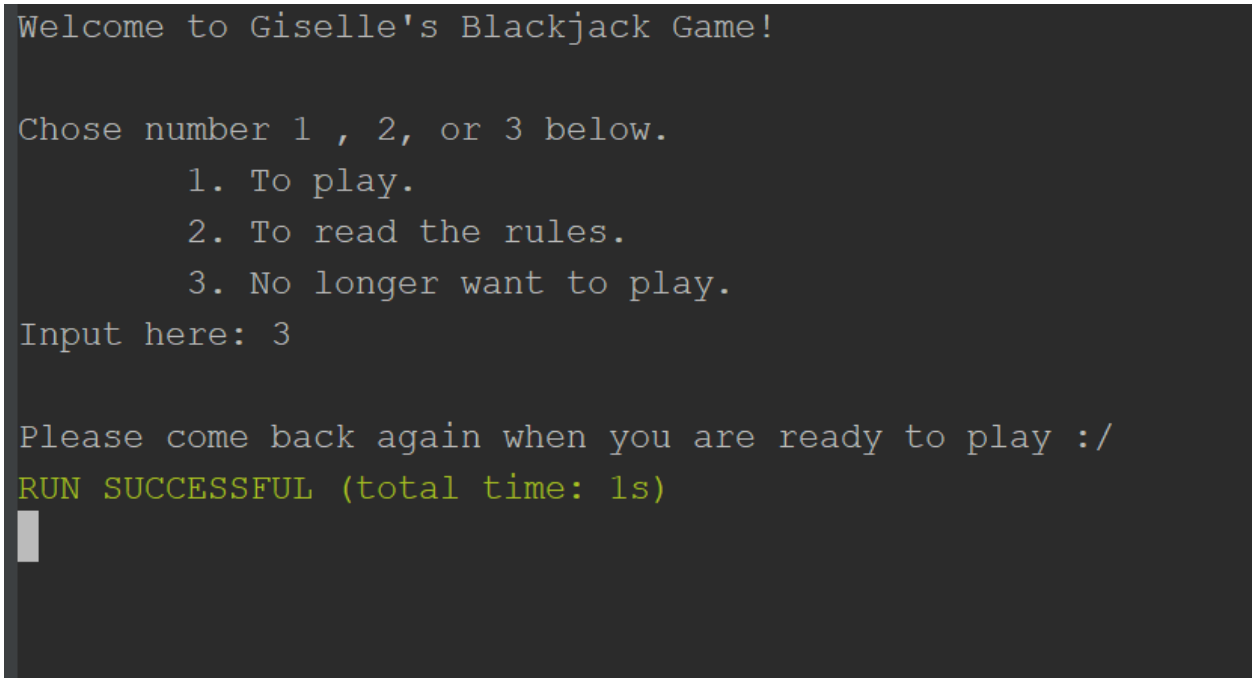
Game ends

Running Screenshots

```
Welcome to Giselle's Blackjack Game!

Chose number 1 , 2, or 3 below.
        1. To play.
        2. To read the rules.
        3. No longer want to play.
Input here: 1

BEAT THE DEALER!

Insert player name here: Giselle
How much are you withdrawing today?
Input: $500
How much would you like to bet? Must be at least $1 and at most $500.
Input: $50

Dealer's hand: 9
Giselle's hand: 10  2  = 12.
Do you want to hit(h) or stand(s)?h

Dealer's hand: 9  6  = 15

Giselle's new card is: 2
Total is now = 14
```

```
Do you want to hit y or n?y

Giselle's new card is: 4
Total is now = 18

Do you want to hit y or n?n

Dealer's hand: 9  6  = 15
Giselle's total: 18

Dealer's new card is: 4
Total is now = 19
Giselle's total: 18

Giselle lost!
Dealer wins!

Giselle's new total: 450
RUN SUCCESSFUL (total time: 22s)
```

```
Welcome to Giselle's Blackjack Game!

Chose number 1 , 2, or 3 below.
        1. To play.
        2. To read the rules.
        3. No longer want to play.
Input here: 2

BLACKJACK : BEAT THE DEALER


How to win!
1. The main goal of blackjack is to beat the dealer at a higher value without going over 21(AKA without going bust).
2. You can also win if the dealer busts and you don't.

Card Values:
1. Deuces through nines are all counted at face value.
2. Tens, jacks, queens, and kings have a value of ten.
3. Aces can either be a 1 or 11.

Rules:
1. Each hand begins with giving everyone two cards.
2. All cards are dealt face up except for the dealers last card.
3. Players act first with two options: Hitting(Tap), Standing(Wave)

        Hitting: You want another card so you tap the table.
        Players can tap as many times as they want until they stand(are happy with their hand) or when they go bust.

        Standing: You wave to show that you are happy with you hand, therefore will not another card.


When your ready to play input 1.
Input:
```



Program

```
#include <cstdlib>
#include <iostream>
#include <iomanip>
#include <string>
#include <cmath>
#include <ctime>

using namespace std;

void Intro();
void diaRuls();

int main(int argc, char** argv)
{
   srand(time(0));                                          //random

   const int MIN_BET = 1;                                       //Player can not bet before 1

   char choose;                                             //choosing between the options in
the switch menu using char
   char move;
```

```
   char agree;                                                    //Player's agreement hit or not
   string name;                                                   //receive and display user's name

   float mony = 0;                                                //Players money
   float bet = 0;                                                 //bet made

   int dealC, dealH, carP1, carP2, totP, totD, newCard;                    //Dealer's cards,
Player's cards, Totals in both



  Intro();


   do
   {
       cout << "Input here: ";
       cin >> choose;
       switch(choose)
       {
          case '2':
          {
             cout << endl << "BLACKJACK : BEAT THE DEALER" << endl << endl;
//display game rules


             cout <<  endl << "How to win!" << endl << "1. The main goal of blackjack is to beat "
//the two ways to win
                   "the dealer at a higher value without going over 21(AKA without going bust)."
                   << endl << "2. You can also win if the dealer busts and you don't." << endl;

             cout << endl << "Card Values:" << endl;                                  //card
values

             cout << "1. Deuces through nines are all counted at face value." << endl
                   << "2. Tens, jacks, queens, and kings have a value of ten." << endl
                   << "3. Aces can either be a 1 or 11." << endl;

             cout <<  endl << "Rules:" << endl;
//rules before winning
             cout << "1. Each hand begins with giving everyone two cards." << endl <<
                   "2. All cards are dealt face up except for the dealers last card." <<endl <<
                   "3. Players act first with two options: Hitting(Tap), Standing(Wave)"
```

```
                << endl << endl <<'\t' <<"Hitting: You want another card so you tap the
table.\n\tPlayers can tap as many times as they "
                "want until they stand(are happy with their hand) or when they go bust." << endl
                << endl <<'\t' << "Standing: You wave to show that you are happy with you hand,
therefore will not another card."
                << endl;

            do{
            cout << endl << endl << "When your ready to play input 1." << endl << "Input: ";
//to exit rules to play
            cin >> choose;
            } while(!(choose == '1'));                                          //if its not
1 it will re ask to input

            if (choose == '1')                                                  //player
chooses 1 it will then ask money and betting numbers
            {
                cout << endl << "BEAT THE DEALER!" << endl << endl;
                cout<< "Insert player name here: ";
                cin >> name;
                cout << "How much are you withdrawing today?" << endl << "Input: $";
                cin >> mony;
            do
            {

               cout << "How much would you like to bet? Must be at least " << MIN_BET <<  " at
most " << mony << "." << endl;        //betting can be at least 1 or at most their total money
               cout << "Input: $";
               cin >> bet;

               cout << endl;
            }
               while(!(bet >= MIN_BET && bet <= mony));
            }


            break;
          }

           case '1':
           {
              cout << endl << "BEAT THE DEALER!" << endl << endl;              //Player chooses 1
from menu
              cout<< "Insert player name here: ";
```

```
            cin >> name;
            cout << "How much are you withdrawing today?" << endl << "Input: $";
            cin >> mony;
          do
          {

            cout << "How much would you like to bet? Must be at least $" << MIN_BET <<  "
and at most $" << mony << "." << endl;  //betting can be at least 1 or at most their total money
            cout << "Input: $";
            cin >> bet;

            cout << endl;
          }
            while(!(bet >= MIN_BET && bet <= mony));
            break;
        }

        case '3':
        {
          cout << endl << "Please come back again when you are ready to play :/";
          break;
        }
    }
  }while(!(choose == '3' || choose == '2' || choose == '1'));


    if (choose == '1' || choose == '2')                          //game begins
    {
     cout << "Dealer's hand: ";                                  //dealer's cards
     dealC = rand() % 10 + 1;
     dealH = rand() % 10 + 1;
     totD = dealC + dealH;
     cout  << dealC << endl;

     carP1 = rand() % 10 + 1;
     carP2 = rand() % 10 + 1;
     totP = carP1 + carP2;


     cout << name << "'s hand: " << carP1 << "  " << carP2 << "  =";          //players cards
     cout << " " << totP << "." << endl;

     cout << "Do you want to hit(h) or stand(s)?";                  //players choice to hit or
not for a new card
```

```
        cin >> move;

        cout << endl;

        if (move == 'h' || move == 'H')                                  //if player chooses hit
        {
            cout << "Dealer's hand: " << dealC << "  " << dealH << "  = " << totD;
            cout << endl;

            do                                                           //give choice of hitting again until player
says no and or players total is greater than 21 or equal to 21
            {

                newCard = rand() % 10 + 1;
                totP = totP + newCard;

                cout << endl << name << "'s new card is: "<< newCard << endl <<"Total is now = " <<
totP << endl;


                if(totP > 21)
                {
                    cout << "Sorry, you busted! You went over 21, you lose!";              //player goes
over 21, loses and exits game

                    mony = mony - bet;
                    cout << endl << endl << "Your new total: $" << mony;

                    exit(0);
                }

                if(totP == 21)
                {
                    cout << endl << "Congrats!" << endl <<  name << " wins!";               //player get
21 then they win!!
                    mony = mony + bet;
                    cout << endl << endl << name << "'s new total: " << mony ;

                    exit(0);
                }

                cout << endl << "Do you want to hit y or n?";                //based on response will
reloop but if n then onto next section of code
                cin >> agree;
```

```
        } while (!(agree == 'n' || agree == 'N'));



        cout << endl  << "Dealer's hand: " << dealC << "  " << dealH << "  = " << totD;
//player finally chose n
        cout << endl << name << "'s total: " << totP << endl;

         do                                                    //dealers turn to get cards,
but if its right under 17 then it will stop
         {
            newCard = rand() % 10 + 1;
            totD = totD + newCard;


            cout  << endl <<"Dealer's new card is: " << newCard << endl <<"Total is now = " <<
totD << endl;

            cout << name << "'s total: " << totP << endl;

        } while(totD < 17 && totP < 21 );

        if (totD == 21 && !(totP == 21))                          //determining winners
        {
            cout << endl << name <<" lost!" << endl << "Dealer wins!";          // dealer wins if
player does not have 21
            mony = mony - bet;                                    //calculates playere new sum
of money amount
            cout << endl << endl << name << "'s new total: "<< mony;
        }

        if(totD > 21)                                             //dealers amount over 21 = busted
- player wins!
        {
            cout << endl << "Dealer busted!";
            cout << endl << "Congrats!" << endl <<  name << " wins!";
            mony = mony + bet;
            cout << endl << endl << name << "'s new total: " << mony ;
        }
```

```
        else if (totP < totD)                                   //dealer total is great then
players, player loses!
        {
           cout << endl << name <<" lost!" << endl << "Dealer wins!";
           mony = mony - bet;
           cout << endl << endl << name << "'s new total: "<< mony;

        }
        else if ( totP > totD)                                  //player total is great then
dealers, player wins!
        {
            cout << endl << "Congrats!" << endl <<  name << " wins!";
           mony = mony + bet;
           cout << endl << endl << name << "'s new total: " << mony ;
        }
        else if (totP == totD)                                  //if values the same then its a
tie
        {
           cout << endl << "A Tie!" << endl << "No Winner :(";

        }

      }

      if (move == 's' || move == 'S')                           //player chooses to stay
      {
        cout << "Dealer's hand: " << dealC << "  " << dealH << "  = " << totD;
        cout << endl;

        cout << name << "'s hand: " << carP1 << "  " << carP2 << "  = " << totP << endl;

        do
         {
           newCard = rand() % 10 + 1;                           //Dealers turn to hit but
only if under 17 then its stops
           totD = totD + newCard;


           cout  << endl <<"Dealer's new card is: " << newCard << endl <<"Total is now = " <<
totD << endl;

           cout << name << "'s total: " << totP << endl;

        } while(totD < 17 && totP < 21 );
```

```
        if (totD == 21 && !(totP == 21))                                    //determining winners
        {
            cout << endl << name <<" lost!" << endl << "Dealer wins!";              // dealer wins if
player does not have 21
            mony = mony - bet;                                              //calculates playere new sum
of money amount
            cout << endl << endl << name << "'s new total: "<< mony;
        }

        if(totD > 21)                                                       //dealers amount over 21 = busted
- player wins!
        {
            cout << endl << "Dealer busted!";
            cout << endl << "Congrats!" << endl <<  name << " wins!";
            mony = mony + bet;
            cout << endl << endl << name << "'s new total: " << mony ;
        }


         else if (totP < totD)                                              //dealer total is great then
players, player loses!
        {
            cout << endl << name <<" lost!" << endl << "Dealer wins!";
            mony = mony - bet;
            cout << endl << endl << name << "'s new total: "<< mony;

        }

         else if ( totP > totD)                                             //player total is great then
dealers, player wins!
        {
             cout << endl << "Congrats!" << endl <<  name << " wins!";
            mony = mony + bet;
            cout << endl << endl << name << "'s new total: " << mony ;
        }

         else if (totP == totD)                                             //if values the same then its a
tie
        {
            cout << endl << "A Tie!" << endl << "No Winner :(";

        }
```

```
        }



    }




    return 0;
}

void Intro()
    {
        cout << "Welcome to Giselle's Blackjack Game!" << endl << endl;
//Introduction menu
            cout << "Chose number 1 , 2, or 3 below." << endl << "\t1. To play." <<
            endl << "\t2. To read the rules." << endl << "\t3. No longer want to play." << endl;
    }
```