Project 1
<BlackJack>

CIS-17-A 48096
Name: Jose Morales
Date: 10/28/16

Introduction

-BlackJack

This version of blackjack that I made is quite different from the commonly know blackjack games. In this version of blackjack, the player is playing solely against the computer to see who can get the highest hand without having to bust. This game consists of the user gaining two random cards at the beginning of the game. The cards mentioned are valued differently, the ace is valued as a one but can be given the value of eleven depending on what value the user picks. The rest of the cards have a value between two-ten. When the computer or the user receives a king, queen, or jack it automatically gets the value of ten. Cards two-nine receive their corresponding face value. After the player has revived his two cards, the computer will generate two random cards too;however, the computer only shows one card face up and hides the other card away from the player. After these interactions, the user is given the choice to "hit". This option will give another random card to the player to add to his hand. The user can also choose the "stand" option in order to stop receiving cards and play with those he has in hand. The player will be given the chance to "hit" as long as his hand total is below twenty-one. Once the player decides to "stand" both the dealer's hand and the player's hands are compared. There are different scenarios when comparing . If one hand goes bust than the other hand is giving the win, if no hand has gone bust than the one closest to twenty-one wins the game. The last choice is given when both hands are equal to each other. In this instance it is classified as a "push" or a tie.

Summary

Project Size:474 lines
Number of Variables: 24
Functions:12

While doing this project I was able to add various concepts from the previous csc class more easily and smoothly. I managed to clean this code out and organize it way more so the user or person looking at the code can understand what is on screen. I tried to use various function to make the program more effective by not writing down the same code over and over again. I really did enjoy working on this project because i was able to use concepts i've learned about more smoothly and incorporate in concepts into this project. I do confess, i did have some difficult times while working. I would try to to use a code one one , but it was not effective. I did manage to learn from these mistakes throughout the project. Some concepts seemed like i could not incorporate or simply I didn't see a way according to my thought process. I would really like to improve this project by using the classes after I get more hang of them. Furthermore, it would be awesome to add a system of money with the option to split or double down. It took a while to think about how to input all the new concepts into this project which helped in the end.

Description

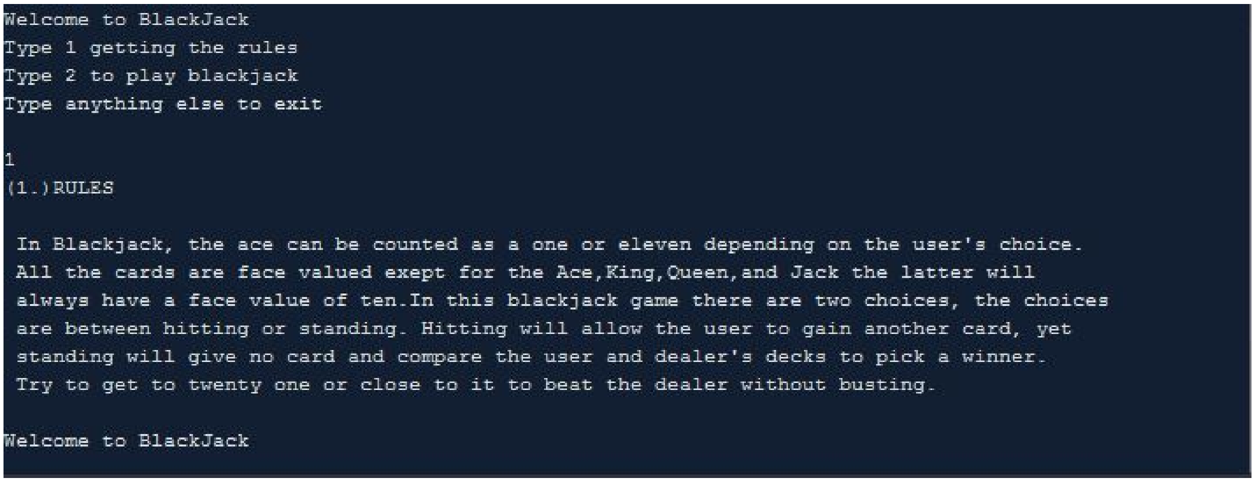
The main objective of doing this blackjack game was to show my ability of using previous concepts from last class and in addition adding new concepts from new chapters to be able to show my understanding in making this game with sufficient logic while using concepts.

Sample I/O- Since this program uses a random number generator for cards, the numbers in the jpgs will not be the same.

Start of Program

When the user starts the program a do-while loop is called on a function “mean();”. The reasoning behind this function is to let the user choice between playing blackjack or reading the rules in case one is not familiar with how the game is played.

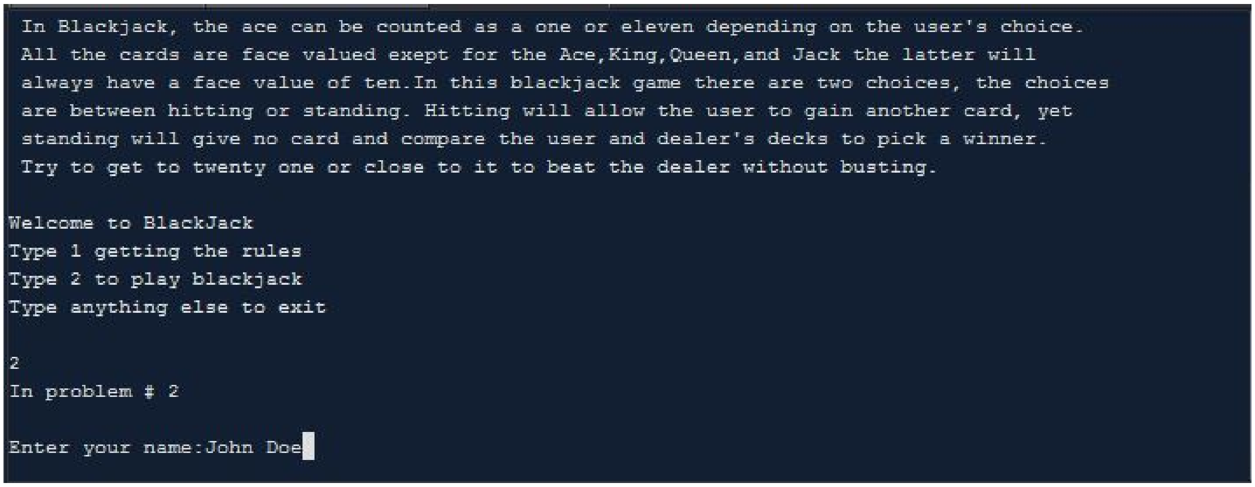
In the JPG below, it shows the user being greeted by the program and giving them an option of reading the rules by entering “1” into the menu.



```
Welcome to BlackJack
Type 1 getting the rules
Type 2 to play blackjack
Type anything else to exit

1
(1.)RULES

 In Blackjack, the ace can be counted as a one or eleven depending on the user's choice.
 All the cards are face valued exept for the Ace,King,Queen,and Jack the latter will
 always have a face value of ten.In this blackjack game there are two choices, the choices
 are between hitting or standing. Hitting will allow the user to gain another card, yet
 standing will give no card and compare the user and dealer's decks to pick a winner.
 Try to get to twenty one or close to it to beat the dealer without busting.

Welcome to BlackJack
```

When the player enters “1” into the program function “read();” is called out. This function reads an array of strings from a “read.txt” doing this allows less lines and clutter in order for more organization and showing a new concept.

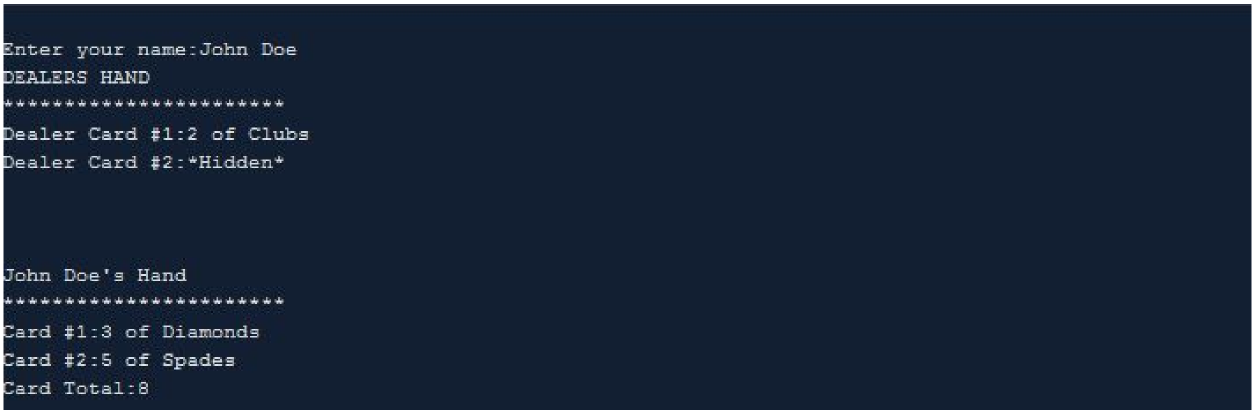
As the reader enters “1” to read the rules, the program loops back to the menu in order to choice to play the game. Once the player enters “2” the user is asked to enter their name so the program can assign their name to their hand total.



```
 In Blackjack, the ace can be counted as a one or eleven depending on the user's choice.
 All the cards are face valued exept for the Ace,King,Queen,and Jack the latter will
 always have a face value of ten.In this blackjack game there are two choices, the choices
 are between hitting or standing. Hitting will allow the user to gain another card, yet
 standing will give no card and compare the user and dealer's decks to pick a winner.
 Try to get to twenty one or close to it to beat the dealer without busting.

Welcome to BlackJack
Type 1 getting the rules
Type 2 to play blackjack
Type anything else to exit

2
In problem # 2

Enter your name:John Doe
```

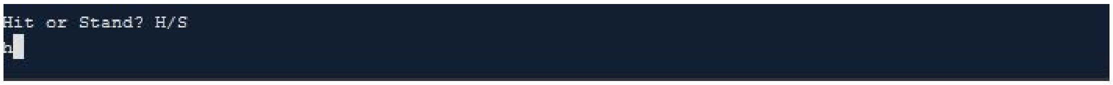
After the user has entered their name the program gives two random cards to the computer/dealer. By doing so, the function “hitValue();” is called down and also function “suit();” is called. The function “hitValue();” will obtain a random number from the range of one to thirteen. After getting this number, a switch(); statement is passed within this function that will give values eleven to thirteen a value of ten because these cards are the jacks,kings, and queens. After this function is done doing this process twice for the dealer’s hand, function “suit();” takes place. It does a similar process, but this time the switch is 4 elements long and it sets the number equal to a string for each suit. If the dealer were to get an ace it would be counted as an automatic eleven unless total of both the cards exceeds eleven then it will be counted as a one. The process will continue as long as the dealer’s hand is equal to no more than seventeen

It's important to note that only one card is face up and the other is hidden away from the user in regards to the dealer’s hand.



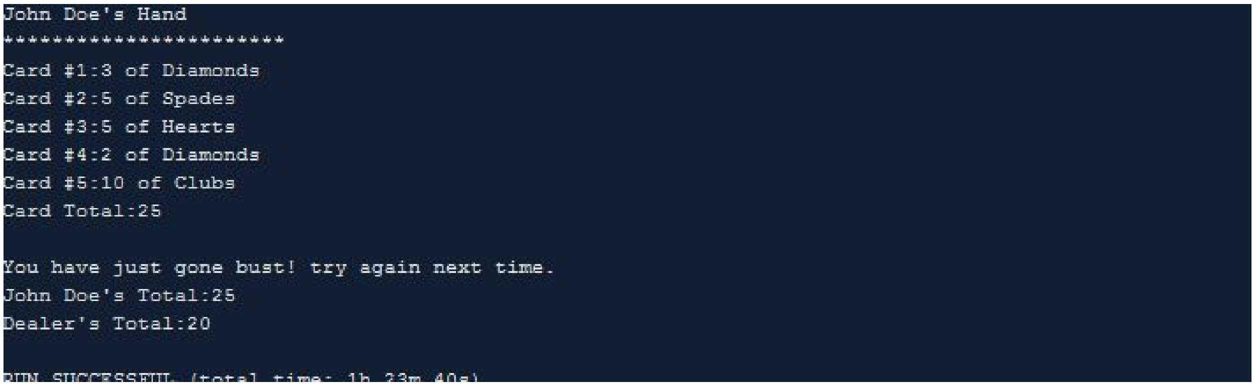
```
Enter your name:John Doe
DEALERS HAND
***********************
Dealer Card #1:2 of Clubs
Dealer Card #2:*Hidden*



John Doe's Hand
***********************
Card #1:3 of Diamonds
Card #2:5 of Spades
Card Total:8
```

The same process happens when the player is given to cards only that in this instance the player has the freedom to choose what value to give the ace in case they get one because of the if nested loops in the program. Both cards are added up for the user. The function “check21();”is called in cause the player has a total of twenty-one. If so, then the player wins. In case this scenario does not happen then the player is allowed to hit as long has their total is less than 21. This will keep on calling the hitValue(); and “suit();” function until done hitting.



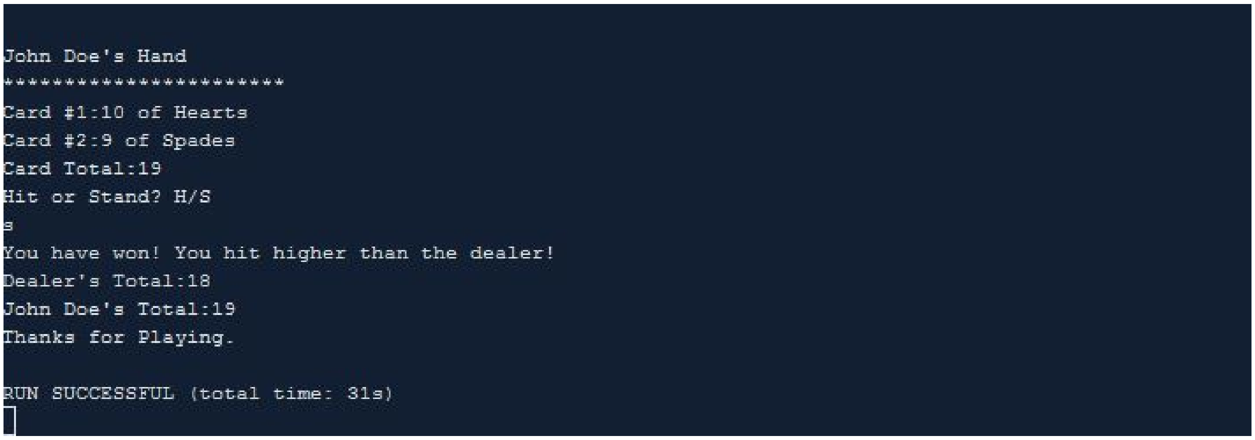
```
Hit or Stand? H/S
h
```

Once the player picks hit then he will be given another card. Functions “hitValue();” and “suit();” will be called once again to set the next card’s value. In the instance below the player decided to hit five times ;however, this time he was not greeted with the “hitValue();” fucntion but with the “bust():” function since the player’s total hand value exceeded more than 21. At this point the blackjack game ends in a game over.



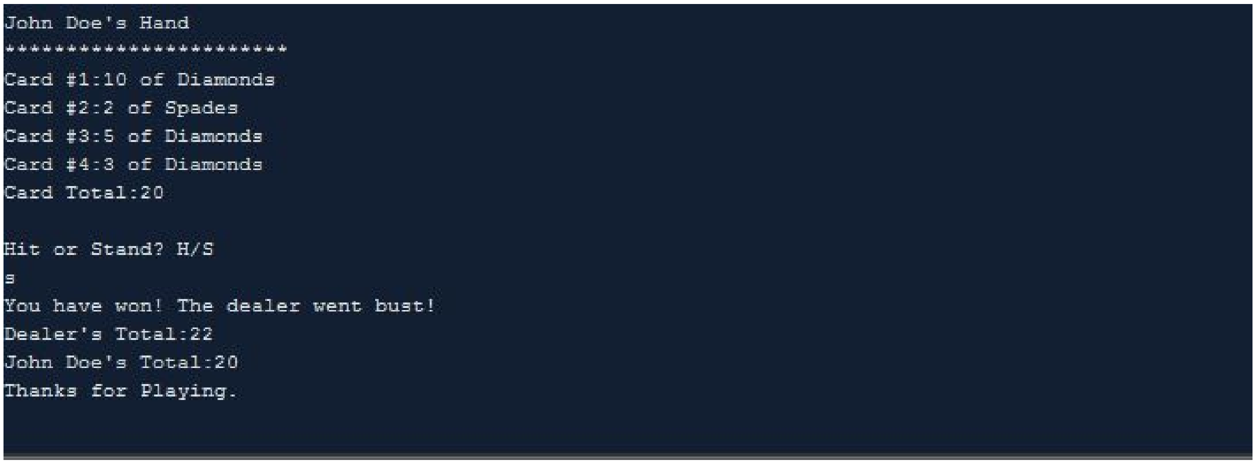
```
John Doe's Hand
***********************
Card #1:3 of Diamonds
Card #2:5 of Spades
Card #3:5 of Hearts
Card #4:2 of Diamonds
Card #5:10 of Clubs
Card Total:25

You have just gone bust! try again next time.
John Doe's Total:25
Dealer's Total:20

RUN SUCCESSFUL (total time: 1h 23m 40s)
```

In the instance below the player decided to stay and not hit after the third card. When the player decides to do this function “compare();” is brought now to compare the total hand’s of both the user and computer/dealer. In the picture below the user won since he had a higher hand than the dealer.



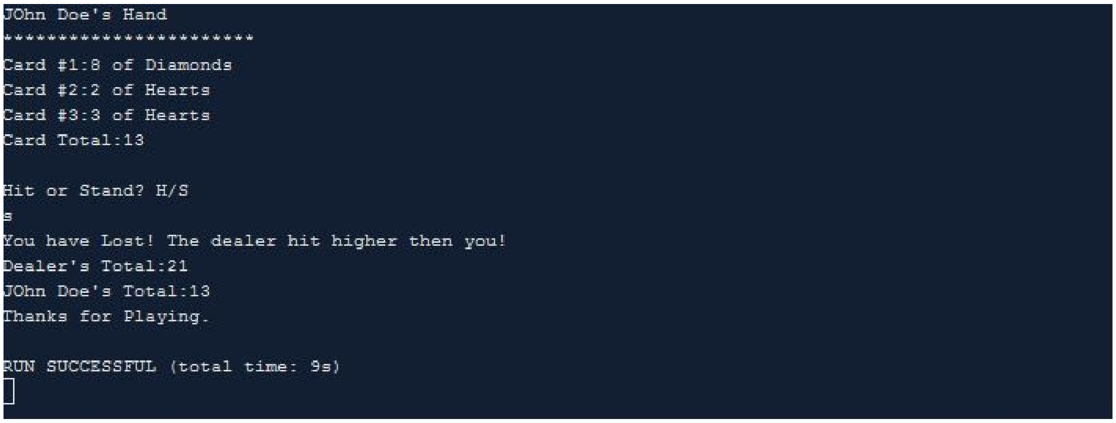
```
John Doe's Hand
***********************
Card #1:10 of Hearts
Card #2:9 of Spades
Card Total:19
Hit or Stand? H/S
s
You have won! You hit higher than the dealer!
Dealer's Total:18
John Doe's Total:19
Thanks for Playing.

RUN SUCCESSFUL (total time: 31s)
```

In this different situation the player decided to stay since he was at a solid twenty. Once again the function “compare():” is brought down. In this Situation the player has won because the dealer went bust with a score of twenty-two. An automatic win for the player.



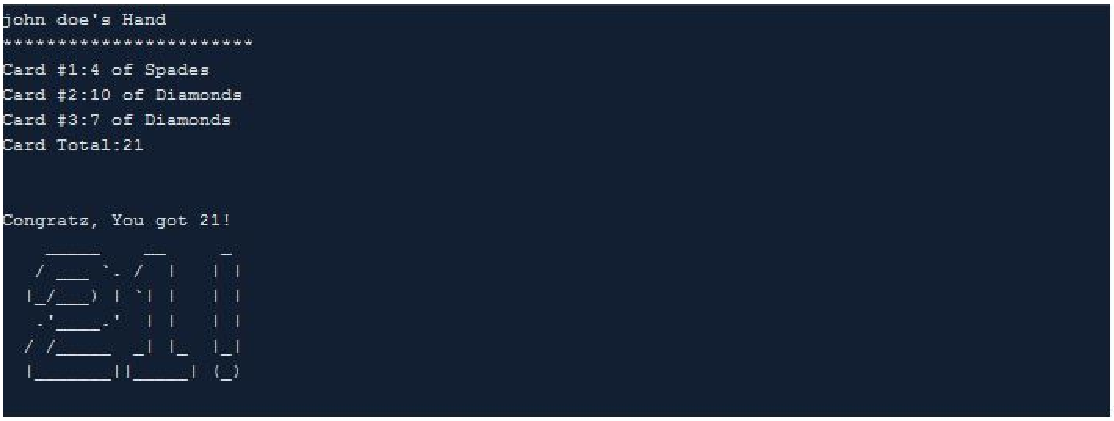
```
John Doe's Hand
***********************
Card #1:10 of Diamonds
Card #2:2 of Spades
Card #3:5 of Diamonds
Card #4:3 of Diamonds
Card Total:20

Hit or Stand? H/S
s
You have won! The dealer went bust!
Dealer's Total:22
John Doe's Total:20
Thanks for Playing.
```

In a different scenario the dealer decides to stay with a low hand value of thirteen. In this instance the function “compare();” gives the win to the dealer since they either had twenty-one or a higher hand than the player.



```
JOhn Doe's Hand
***********************
Card #1:8 of Diamonds
Card #2:2 of Hearts
Card #3:3 of Hearts
Card Total:13

Hit or Stand? H/S
s
You have Lost! The dealer hit higher then you!
Dealer's Total:21
JOhn Doe's Total:13
Thanks for Playing.

RUN SUCCESSFUL (total time: 9s)
```

In this last scenario the player decides to hit three times and gets a hand value of twenty-one. Since he got this score the function ”check21();” is called down and gives the user a win with a display.

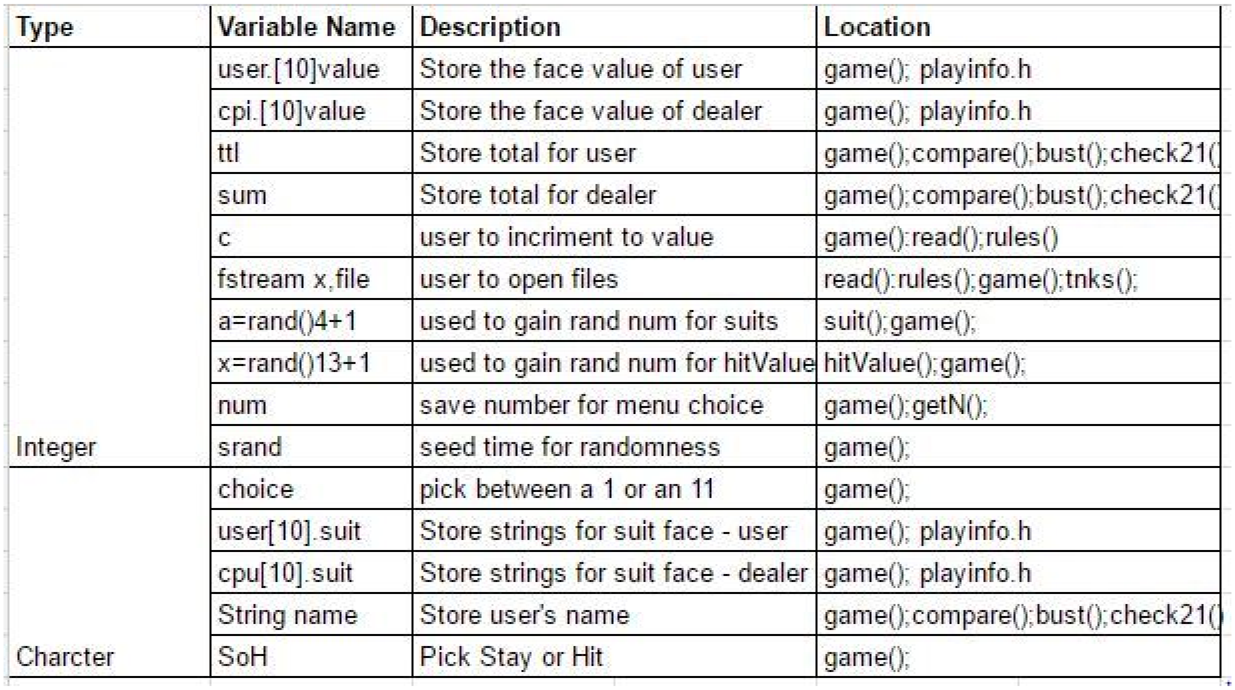


```
john doe's Hand
***********************
Card #1:4 of Spades
Card #2:10 of Diamonds
Card #3:7 of Diamonds
Card Total:21


Congratz, You got 21!
    _____    __     _
   / ___ `. /  |   | |
  |_/___) | `| |   | |
   .'____.'  | |   | |
  / /_____  _| |_  |_|
  |_______||_____| (_)
```

Pseudo Code

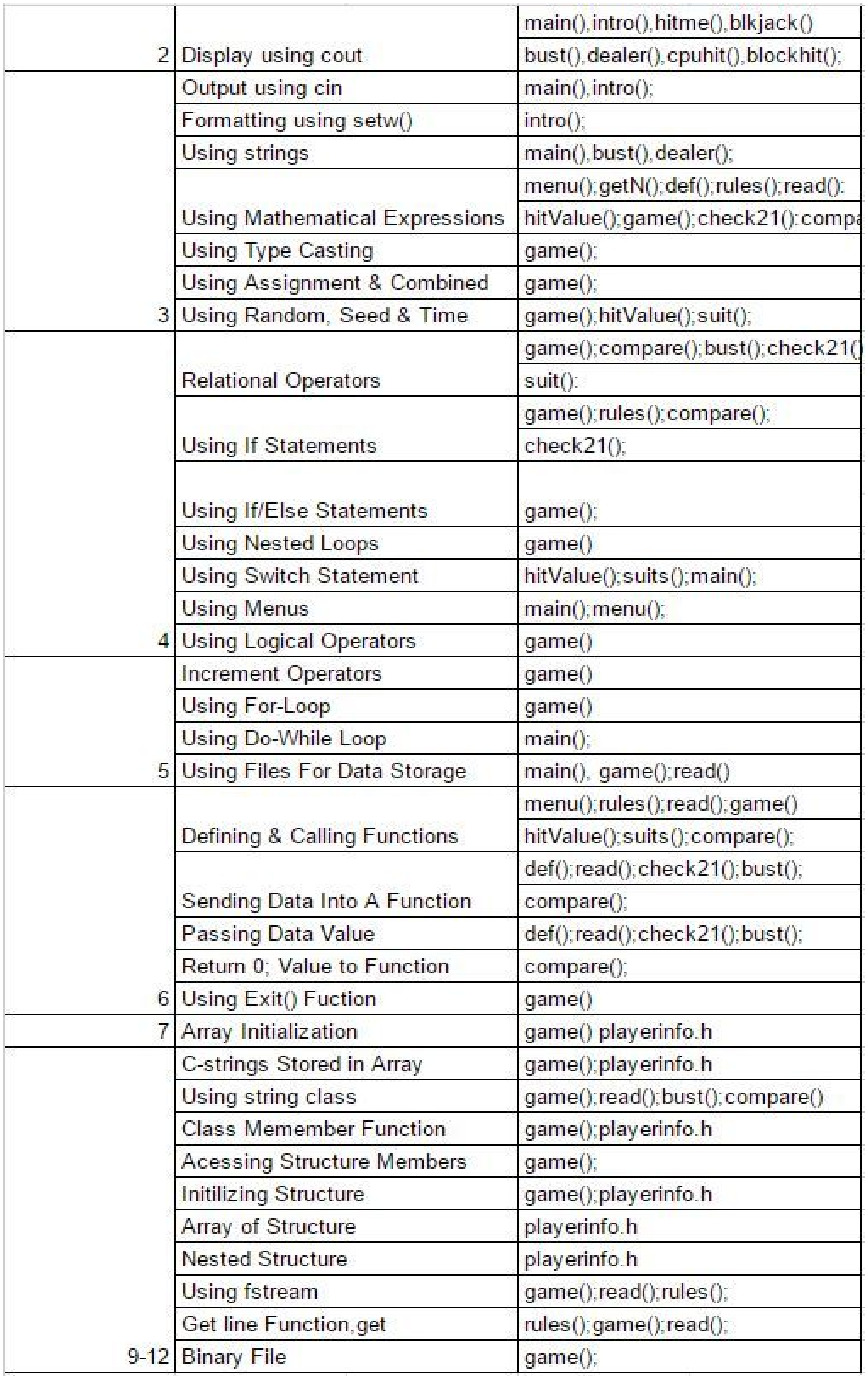
```
Initialize
    Bring Down Menu
        Case 1: rules
        Case 2:game
        Default:exit
    If Case 1:
        Function :read():
            Function brings down text from rules.txt using strings
            Once user is done, loop back to menu.
    If Case 2:
        Function:game();
            Call 2 random cards for both computer and user
            Initialize both cards with suit and face value.
                Computer keeps on hitting until total it equal to or less than 17.
                If computer get an ace set it as 11 at start after that a 1 if total is
                less than 11
            Once its the user's turn
                Check to see if user has a total of 21
                    If 21 then end game since user has won
                Check if user wants to hit or stand
                    If user picks hit
                        Call function to give another random card
                            Add up random cards.
                    If user picks stand
                        Call function to compare value
                            If user value is more than 21
                                User went bust
                            If user value less than 21 but less than cpu
                                User lost
                            If user value more than cpu
                                User won
                            If user value = cpu
                                Tie
                    Repeat above if user has opportunity to hit or stand
```

Variables



Type	Variable Name	Description	Location
Integer	user.[10]value	Store the face value of user	game(); playinfo.h
	cpi.[10]value	Store the face value of dealer	game(); playinfo.h
	ttl	Store total for user	game();compare();bust();check21(
	sum	Store total for dealer	game();compare();bust();check21(
	c	user to incriment to value	game():read();rules()
	fstream x,file	user to open files	read():rules();game();tnks();
	a=rand()4+1	used to gain rand num for suits	suit();game();
	x=rand()13+1	used to gain rand num for hitValue	hitValue();game();
	num	save number for menu choice	game();getN();
	srand	seed time for randomness	game();
Charcter	choice	pick between a 1 or an 11	game();
	user[10].suit	Store strings for suit face - user	game(); playinfo.h
	cpu[10].suit	Store strings for suit face - dealer	game(); playinfo.h
	String name	Store user's name	game();compare();bust();check21()
	SoH	Pick Stay or Hit	game();

Constructs



2	Display using cout	main(),intro(),hitme(),blkjack()
		bust(),dealer(),cpuhit(),blockhit();

3	Output using cin	main(),intro();
	Formatting using setw()	intro();
	Using strings	main(),bust(),dealer();
	Using Mathematical Expressions	menu();getN();def();rules();read(): hitValue();game();check21():compa
	Using Type Casting	game();
	Using Assignment & Combined	game();
	Using Random, Seed & Time	game();hitValue();suit();
4	Relational Operators	game();compare();bust();check21(suit():

	Using If Statements	game();rules();compare();
		check21();

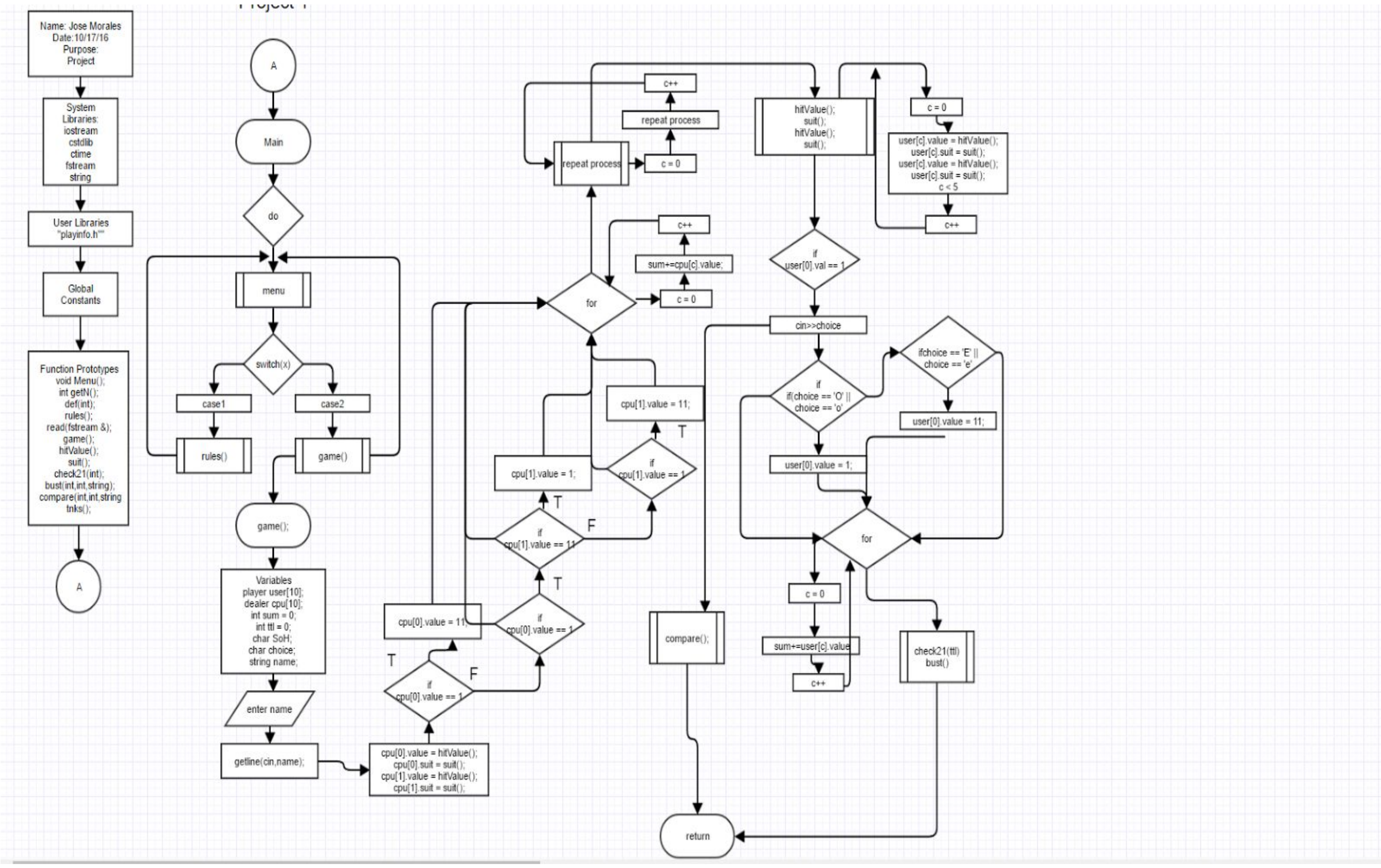
	Using If/Else Statements	game();
	Using Nested Loops	game()
	Using Switch Statement	hitValue();suits();main();
	Using Menus	main();menu();
	Using Logical Operators	game()
5	Increment Operators	game()
	Using For-Loop	game()
	Using Do-While Loop	main();
	Using Files For Data Storage	main(), game();read()

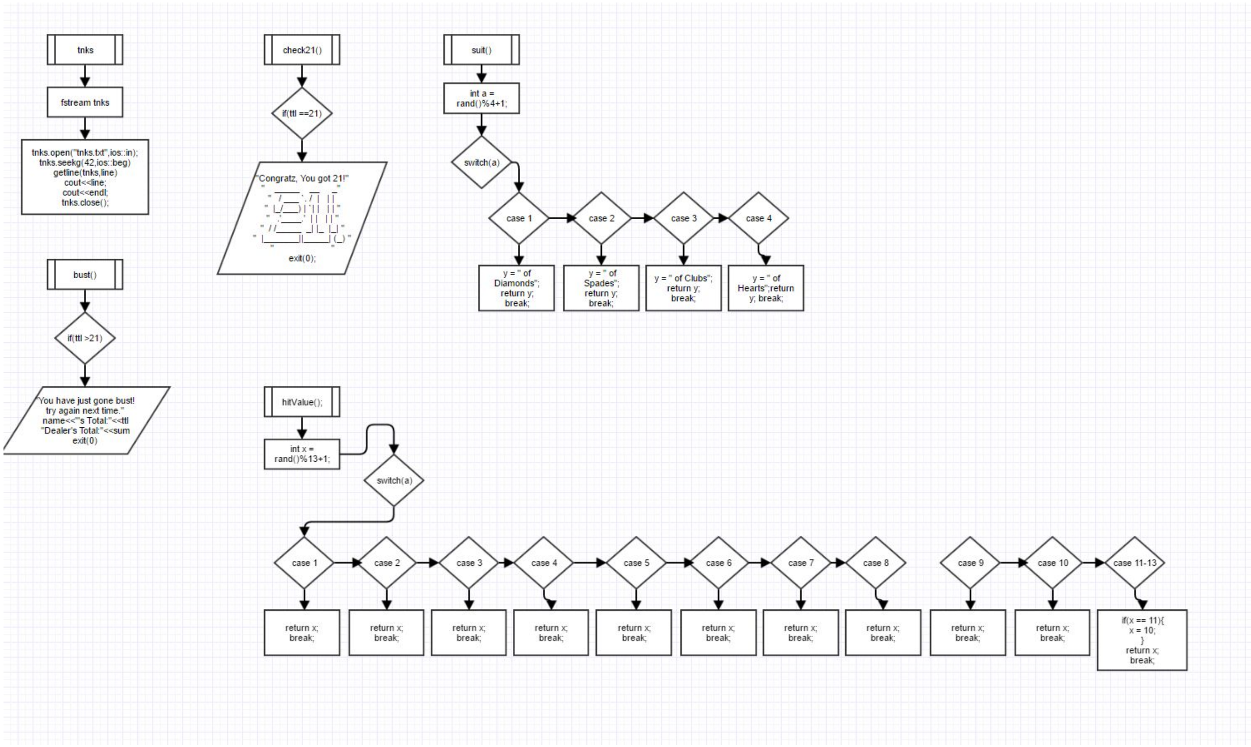
6	Defining & Calling Functions	menu();rules();read();game()
		hitValue();suits();compare();

	Sending Data Into A Function	def();read();check21();bust();
		compare();

	Passing Data Value	def();read();check21();bust();
	Return 0; Value to Function	compare();
	Using Exit() Fuction	game()
7	Array Initialization	game() playerinfo.h
9-12	C-strings Stored in Array	game();playerinfo.h
	Using string class	game();read();bust();compare()
	Class Memember Function	game();playerinfo.h
	Acessing Structure Members	game();
	Initilizing Structure	game();playerinfo.h
	Array of Structure	playerinfo.h
	Nested Structure	playerinfo.h
	Using fstream	game();read();rules();
	Get line Function,get	rules();game();read();
	Binary File	game();

Flowcharts


Project 1
Name: Jose Morales
Date:10/17/16
Purpose:
Project
System
Libraries:
iostream
cstdlib
ctime
fstream
string
User Libraries
"playinfo.h""
Global
Constants
Function Prototypes
void Menu();
int getN();
def(int);
rules();
read(fstream &);
game();
hitValue();
suit();
check21(int);
bust(int,int,string);
compare(int,int,string
tnks();
A
A
Main
do
menu
switch(x)
case1
case2
rules()
game()
game();
Variables
player user[10];
dealer cpu[10];
int sum = 0;
int ttl = 0;
char SoH;
char choice;
string name;
enter name
getline(cin,name);
cpu[0].value = hitValue();
cpu[0].suit = suit();
cpu[1].value = hitValue();
cpu[1].suit = suit();
if
cpu[0].value == 1
T
F
cpu[0].value = 11;
if
cpu[0].value == 1
T
if
cpu[1].value == 11
T
F
cpu[1].value = 1;
if
cpu[1].value == 1
T
cpu[1].value = 11;
for
c = 0
sum+=cpu[c].value;
c++
repeat process
c = 0
repeat process
c++
hitValue();
suit();
hitValue();
suit();
c = 0
user[c].value = hitValue();
user[c].suit = suit();
user[c].value = hitValue();
user[c].suit = suit();
c < 5
c++
if
user[0].val == 1
cin>>choice
if
if(choice == 'O' ||
choice == 'o'
ifchoice == 'E' ||
choice == 'e'
user[0].value = 11;
user[0].value = 1;
for
c = 0
sum+=user[c].value
c++
compare();
check21(ttl)
bust()
return



Flowcharts included in file if hard to see

Code

```
/*
 * File: Project
 * Author: Jose Morales
 * Purpose:
 * Due October 28, 2016, 11:59 PM
 */

///System Libraries
#include <iostream>
#include <cstdlib>
#include <ctime>
#include <fstream>
#include <string>
using namespace std;

//User Libraries
#include "playinfo.h"
//Global Constants

//Function Prototypes
void Menu();
int getN();
void def(int);
void rules();
void read(fstream &);
void game();
int hitValue();
string suit();
void check21(int);
void bust(int,int,string);
void compare(int,int,string);
void tnks();
//Begin Execution Here!!!
int main(int argv,char *argc[]){
    int num;
    do{
        Menu();
        num=getN();
        switch(num){
        case 1:    rules();break;
        case 2:    game();break;
        default:   def(num);}
    }while(num>=1&&num<=2);
    return 0;
}

void Menu(){
    cout<<"Welcome to BlackJack"<<endl;
    cout<<"Type 1 getting the rules"<<endl;
    cout<<"Type 2 to play blackjack"<<endl;
    cout<<"Type anything else to exit \n"<<endl;
}

int getN(){
        int num;
        cin>>num;
        cin.ignore();
        return num;
}

void rules(){
    cout<<"(1.)RULES"<<endl<<endl;
    //Declare Variables
        string txtName;
        fstream file;
    //Open File
        file.open("rules.txt", ios::in);
    //Check if file exists
        if(file.fail()){
            cout<<"Error, File not Found!"<<endl;
        }
        else{
            read(file);
        }
        file.close();
```

```
}
//Functions
void read (fstream &file){
   string line;
   int c = 0;
   while(file >> line && c <= 15){
      c++;
      getline(file,line);
      cout<<line<<endl;
   }
   cout<<endl;
}


/*****************************************************************************
**
******************************************************************************
*/
void game(){
   cout<<"In problem # 2"<<endl<<endl;
   //Set the random number seed for variability
   srand(static_cast<unsigned int>(time(0)));
   //Declare Variables
   player user[10];    //Structure for player cards.
   dealer cpu[10];     //Structure for dealer cards.
   int ttl = 0;        //Keep total of user cards.
   int sum = 0;        //Keep total of computer cards
   char SoH;           //Store choice of hitting or standing.
   char choice;        //Choice value of ace.
   string name;        //Store name of user.

   //Input User Name.
   cout<<"Enter your name:";
   getline(cin,name);
   //Computer Blackjack
   cpu[0].value = hitValue();
   cpu[0].suit = suit();
   cpu[1].value = hitValue();
   cpu[1].suit = suit();
```

```
//Check if computer got an ace.
if (cpu[0].value == 1){
    cpu[0].value = 11;
}
if(cpu[1].value == 1){
    if(cpu[1].value == 11){
        cpu[1].value = 1;
    }
    else if(cpu[1].value == 1){
        cpu[1].value = 11;
    }
}
//Check Cards for computer 1 & 2
cout<<"DEALERS HAND"<<endl;
cout<<"************************"<<endl;
cout<<"Dealer Card #1:"<<cpu[0].value<<cpu[0].suit<<endl;
cout<<"Dealer Card #2:*Hidden*"<<endl;
//Check Computer total for 1 & 2.
 for(int c = 0; c < 2; c++){
    sum+=cpu[c].value;
}
//cout<<"Dealer Card Total:"<<sum<<endl;
cout<<endl;
//Check if computer needs a third card.
if(sum <= 17){
    sum = 0;
    cpu[2].value = hitValue();
    if(cpu[2].value == 1){
        if(sum <= 10){
            cpu[2].value == 11;
            sum = 0;
        }
        else if(cpu[2].value == 1){
            cpu[2].value = 1;
            sum = 0;
        }
    }
}
```

```
 //Check Cards for computer 1 & 2 & 3
//cout<<"Dealer Card 1:"<<cpu[0].value<<cpu[0].suit<<endl;
//cout<<"Dealer Card 2:"<<cpu[1].value<<cpu[1].suit<<endl;
//cout<<"Dealer Card 3:"<<cpu[2].value<<cpu[2].suit<<endl;
//sum=0;
//Check Computer total for 1 & 2 & 3.
 for(int c = 0; c < 3; c++){
   sum+=cpu[c].value;
}
//cout<<"Dealer Card Total:"<<sum<<endl;
cout<<endl;


//Check if computer needs a fourth card.
if(sum <= 17){
   sum = 0;
   cpu[3].value = hitValue();
   if(cpu[3].value == 1){
      if(sum <= 10){
         cpu[3].value == 11;
         sum = 0;
      }
      else if(cpu[3].value == 1){
         cpu[3].value = 1;
         sum = 0;
      }
   }
}
 //Check Cards for computer 1 & 2 & 3 & 4
//cout<<"Dealer Card 1:"<<cpu[0].value<<cpu[0].suit<<endl;
//cout<<"Dealer Card 2:"<<cpu[1].value<<cpu[1].suit<<endl;
//cout<<"Dealer Card 3:"<<cpu[2].value<<cpu[2].suit<<endl;
//cout<<"Dealer Card 4:"<<cpu[3].value<<cpu[3].suit<<endl;
sum = 0;
//Check Computer total for 1 & 2 & 3 & 4.
 for(int c = 0; c < 4; c++){
   sum+=cpu[c].value;
}
//cout<<"Card Total:"<<sum<<endl;
```

```
cout<<endl;


//User Blackjack
user[0].value = hitValue(); //First Card
user[0].suit = suit();      //Suit of First Card
user[1].value = hitValue(); //Second Card
user[1].suit = suit();      //Suit of Second Card.

//Show User His Cards.
cout<<name<<"'s Hand"<<endl;
cout<<"***********************"<<endl;
cout<<"Card #1:"<<user[0].value<<user[0].suit<<endl;
cout<<"Card #2:"<<user[1].value<<user[1].suit<<endl;

//Check if player got an ace on first card.
if(user[0].value == 1){
   cout<<"You just got an ace, count as 1 or 11? (O/E)"<<endl;
   cin>>choice;
   if(choice == 'O' || choice == 'o'){
      user[0].value = 1;
   }
   if(choice == 'E' || choice == 'e'){
      user[0].value = 11;
   }
}
//Check if player got an ace on second card.
if(user[1].value == 1){
   cout<<"You just got an ace, count as 1 or 11? (O/E)"<<endl;
   cin>>choice;
   if(choice == 'O' || choice == 'o'){
      user[1].value = 1;
   }
   if(choice == 'E' || choice == 'e'){
      user[1].value = 11;
   }
}
//Add Total for 2 cards.
for(int c = 0; c < 2; c++){
    ttl+=user[c].value;
}
cout<<"Card Total:"<<ttl<<endl;

//Check if user got 21
check21(ttl);
bust(ttl,sum,name);
//Binary from file and out
fstream x;
x.open("binary.dat",ios::out | ios::binary);
x.write(reinterpret_cast<char *>(user),sizeof(user));
x.close();
//Ask for a third card.
if(ttl < 21){
    cout<<"Hit or Stand? H/S"<<endl;
    cin>>SoH;
    if(SoH == 'H' || SoH == 'h'){
        user[2].value = hitValue();
        user[2].suit = suit();
        ttl = 0;
        if(user[2].value == 1){
            cout<<"You just got an ace, count as 1 or 11? (O/E)"<<endl;
            cin>>choice;
            if(choice == 'O' || choice == 'o'){
                user[2].value = 1;
            }
            if(choice == 'E' || choice == 'e'){
                user[2].value = 11;
            }
        }
    }
    //Stand Option - Place Holder
    else{
    compare(ttl,sum,name);
}
//Add Total for 3 cards
for(int c = 0; c < 3; c++){
    ttl+=user[c].value;
}
```

```
//Show User His Cards.
cout<<name<<"'s Hand"<<endl;
cout<<"*********************"<<endl;
cout<<"Card #1:"<<user[0].value<<user[0].suit<<endl;
cout<<"Card #2:"<<user[1].value<<user[1].suit<<endl;
cout<<"Card #3:"<<user[2].value<<user[2].suit<<endl;
//Display total
cout<<"Card Total:"<<ttl<<endl;
cout<<endl;
}
//Check if user got 21
check21(ttl);
bust(ttl,sum,name);
//Ask for a fourth card.
if(ttl < 21){
  cout<<"Hit or Stand? H/S"<<endl;
  cin>>SoH;
  if(SoH == 'H' || SoH == 'h'){
    user[3].value = hitValue();
    user[3].suit = suit();
    ttl = 0;
    if(user[3].value == 1){
      cout<<"You just got an ace, count as 1 or 11? (O/E)"<<endl;
      cin>>choice;
      if(choice == 'O' || choice == 'o'){
        user[3].value = 1;
      }
      if(choice == 'E' || choice == 'e'){
        user[3].value = 11;
      }
    }
  }
  //Stand Option - Place Holder
  else{
  compare(ttl,sum,name);
}
//Add Total for 4 cards
for(int c = 0; c < 5; c++){
  ttl+=user[c].value;
}
//Show User His Cards.
cout<<name<<"'s Hand"<<endl;
cout<<"***********************"<<endl;
cout<<"Card #1:"<<user[0].value<<user[0].suit<<endl;
cout<<"Card #2:"<<user[1].value<<user[1].suit<<endl;
cout<<"Card #3:"<<user[2].value<<user[2].suit<<endl;
cout<<"Card #4:"<<user[3].value<<user[3].suit<<endl;
//Display total
cout<<"Card Total:"<<ttl<<endl;
cout<<endl;
}
//Check if user got 21
check21(ttl);
bust(ttl,sum,name);
//Ask for a fifth card.
if(ttl < 21){
   cout<<"Hit or Stand? H/S"<<endl;
   cin>>SoH;
   if(SoH == 'H' || SoH == 'h'){
      user[4].value = hitValue();
      user[4].suit = suit();
      ttl = 0;
   }
   else{
   compare(ttl,sum,name);
   }
//Add Total for 5 cards
for(int c = 0; c < 6; c++){
   ttl+=user[c].value;
}
//Show User His Cards.
cout<<name<<"'s Hand"<<endl;
cout<<"***********************"<<endl;
cout<<"Card #1:"<<user[0].value<<user[0].suit<<endl;
cout<<"Card #2:"<<user[1].value<<user[1].suit<<endl;
cout<<"Card #3:"<<user[2].value<<user[2].suit<<endl;
cout<<"Card #4:"<<user[3].value<<user[3].suit<<endl;
cout<<"Card #5:"<<user[4].value<<user[4].suit<<endl;
```

```
  //Display total
  cout<<"Card Total:"<<ttl<<endl;
  cout<<endl;
  }
  //Check if user got 21
  check21(ttl);
  bust(ttl,sum,name);
}
//Functions for game
void tnks(){
  fstream tnks;
  string line;
  //Introduce file
    tnks.open("tnks.txt",ios::in);
    tnks.seekg(42,ios::beg);
    getline(tnks,line);
    cout<<line;
    cout<<endl;
  //Close
    tnks.close();

}

void compare(int ttl, int sum, string name){
  if(sum > 21){
    cout<<"You have won! The dealer went bust!"<<endl;
    cout<<"Dealer's Total:"<<sum<<endl;
    cout<<name<<"'s Total:"<<ttl<<endl;
    tnks();
    exit(0);
  }
  if(sum > ttl){
    cout<<"You have Lost! The dealer hit higher then you!"<<endl;
    cout<<"Dealer's Total:"<<sum<<endl;
    cout<<name<<"'s Total:"<<ttl<<endl;
    tnks();
    exit(0);
  }
  if(ttl > sum){
    cout<<"You have won! You hit higher than the dealer!"<<endl;
    cout<<"Dealer's Total:"<<sum<<endl;
    cout<<name<<"'s Total:"<<ttl<<endl;
    tnks();
    exit(0);
  }
  if(ttl = sum){
    cout<<"Game is a tie/push"<<endl;
    cout<<"Dealer's Total:"<<sum<<endl;
    cout<<name<<"'s Total:"<<ttl<<endl;
    tnks();
    exit(0);
  }
}

void bust(int ttl,int sum,string name){
  if(ttl > 21){
    cout<<"You have just gone bust! try again next time."<<endl;
    cout<<name<<"'s Total:"<<ttl<<endl;
    cout<<"Dealer's Total:"<<sum<<endl;
    exit(0);
  }
}

void check21(int ttl){
  if(ttl == 21){
    cout<<endl;
    cout<<"Congratz, You got 21!"<<endl;
    cout<<"    ______    __    _" <<endl;
    cout<<"   / ___ `. /  |   || "<<endl;
    cout<<"  |_/___) | `||   || "<<endl;
    cout<<"   .'____.' ||   || "<<endl;
    cout<<"  / /_____  _| |_  |_| "<<endl;
    cout<<"  |_______||_____| (_) "<<endl;
    cout<<"                         "<<endl;
    exit(0);

  }
}
```

```
string suit(){
    int a = rand()%4+1;
    string y;
    switch(a){
        case 1 : y = " of Diamonds"; return y; break;
        case 2 : y = " of Spades"; return y; break;
        case 3 : y = " of Clubs"; return y; break;
        case 4 : y = " of Hearts";return y; break;
    }
}

int hitValue(){
    int x = rand()%13+1;
    switch(x){
        case 1 : return x; break;
        case 2 : return x; break;
        case 3 : return x; break;
        case 4 : return x; break;
        case 5 : return x; break;
        case 6 : return x; break;
        case 7 : return x; break;
        case 8 : return x; break;
        case 9 : return x; break;
        case 10 : return x; break;
        case 11 :
            if(x == 11){
                x = 10;
            }
            return x;
            break;
        case 12 :
            if(x == 12){
                x = 10;
            }
            return x;
            break;
        case 13 :
            if(x == 13){
                x = 10;
            }
            return x;
            break;
    }
}
/*************************************************************************************
**
**************************************************************************************
*/
void def(int inN){
        cout<<"You typed "<<inN<<" to exit the program"<<endl;
}
```