Project 2
Black Jack

CIS 5 – 44187

Dokyoung Shin

Introduction

This program allows to play the Black Jack easily. This Blackjack game is very simple and enjoyable game to do for everyone and it is the most popular table game.

The goal of Blackjack game is trying to make as close as 21 score. Moreover, the way to win the game is to beat the dealer by getting a higher score than the dealer without going over 21 with any additional cards. Player starts a game with two cards. After displaying player's score and if player's score is less than 21, player can decide to get another card. Other than that, player can hold with player's existing score. If player goes over 21 after getting another card, it will be "Bust." Therefore, this program will automatically display the dealer's cards score and show the winner.

Values of cards:

1. All number cards are worth a number of points equal to their numerical value.

2. In general, Ace is worth either 1 point or 11 points depends on the player's decision. However, in this program, if player's total score is less than 10 points, then program will add 11 points, if not it will add 1 point.

3.Picture cards are worth 10 points.

Kings = 10 points, Queens = 10 points, Jacks = 10 points

This Black Jack game is great for fun and entertainment.

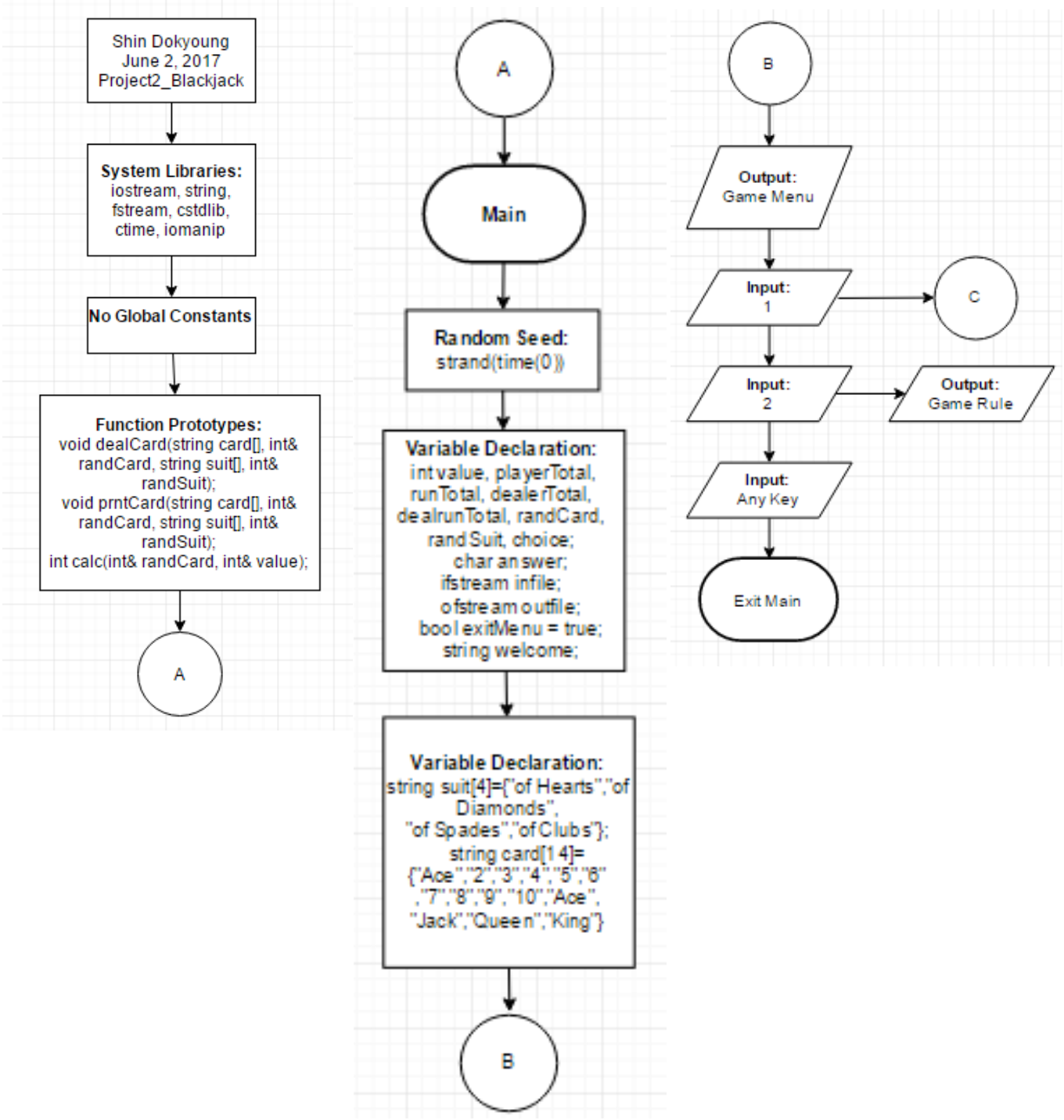
Summary

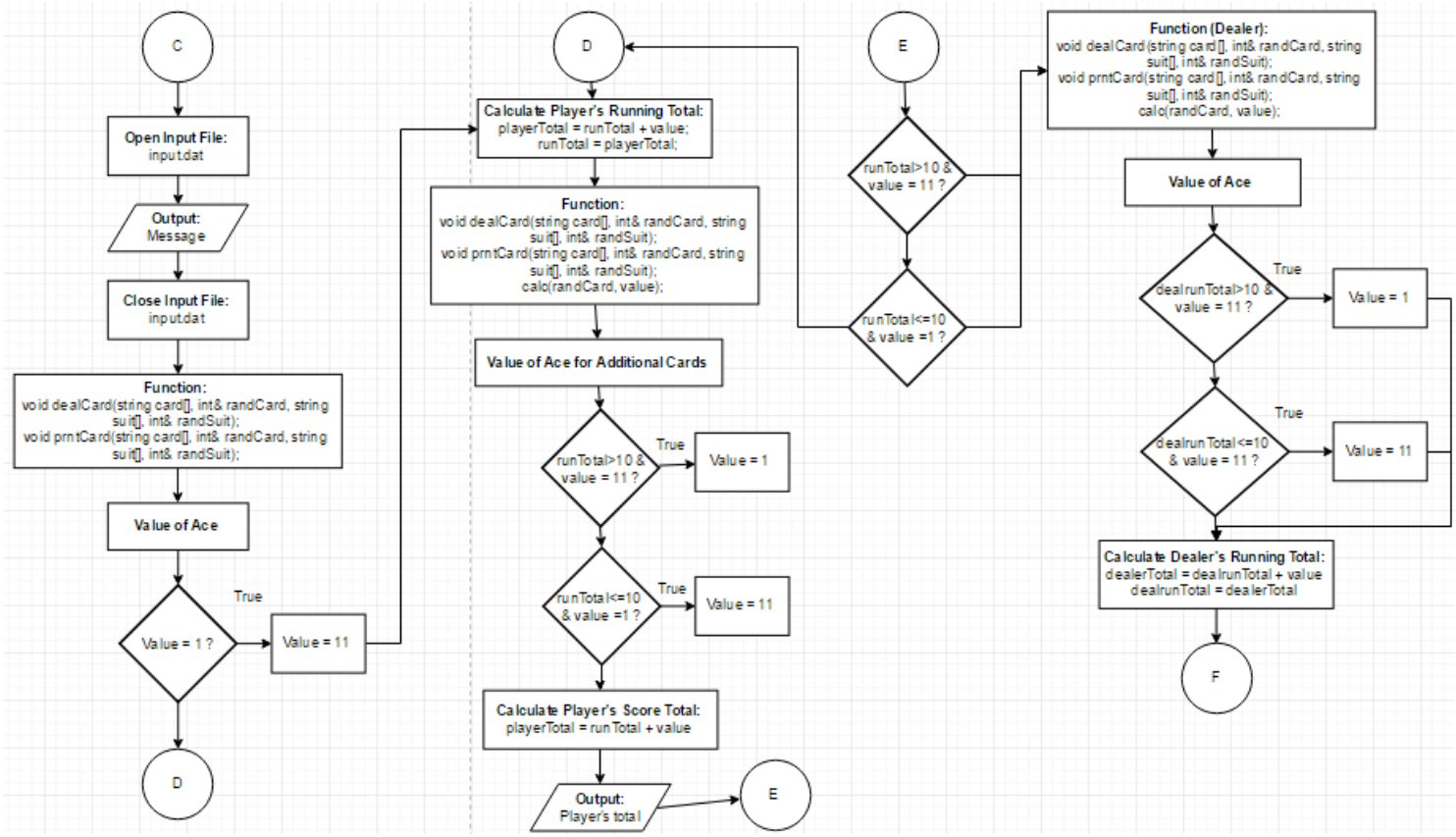
Project size: about 350 lines

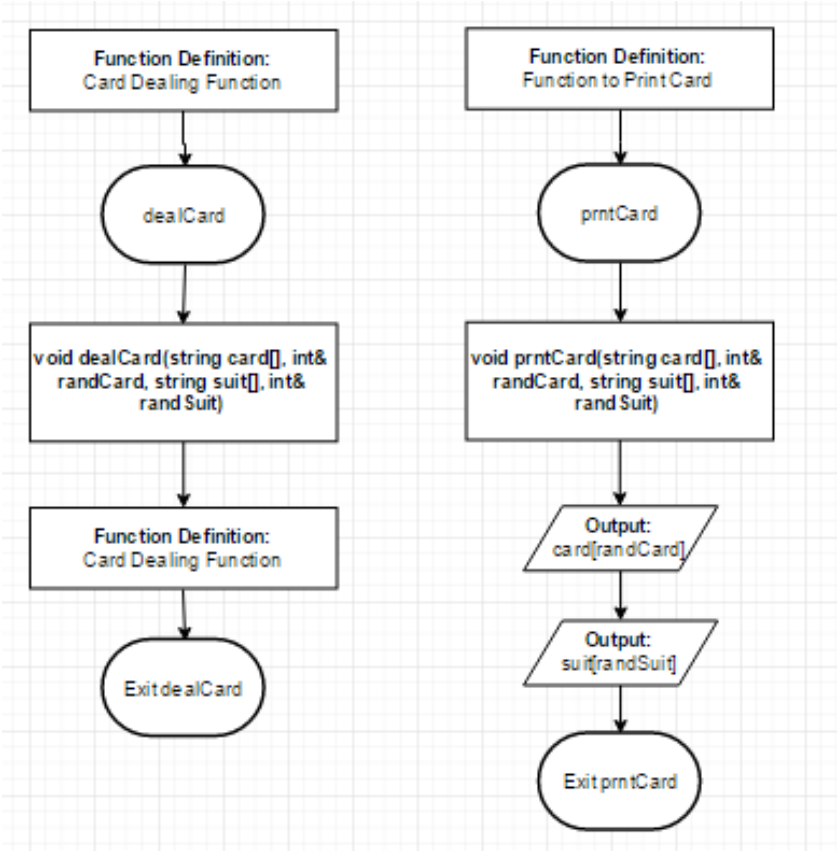
First, I created a simple Blackjack game program.

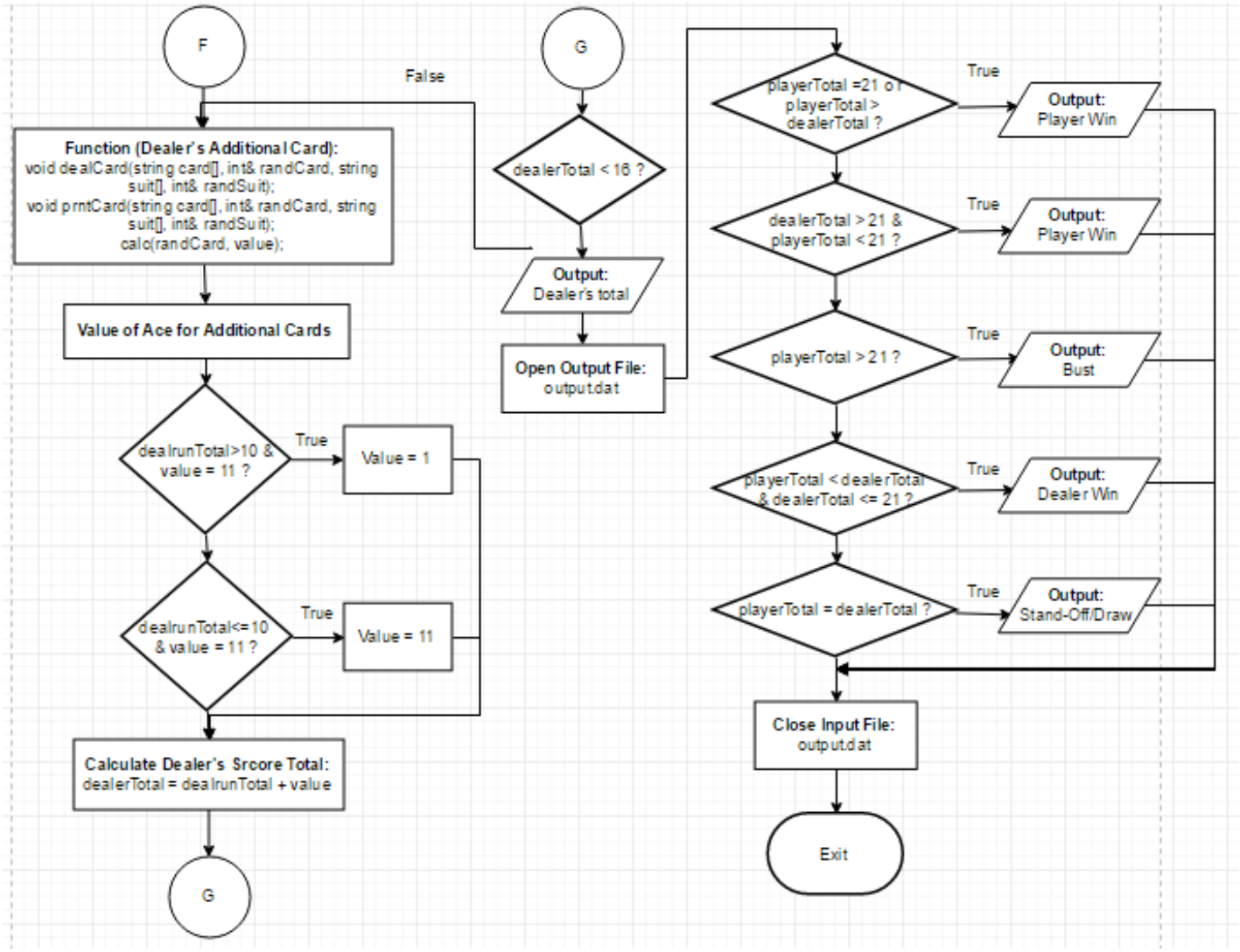
This project includes many concepts that we learned from the chapters in the textbook *Problem Solving with C++*. I realized that C++ has numerous functions and it can make many different ways to create the programs. It was very interesting time to learn and figure it out.

Flow Chart


Shin Dokyoung
June 2, 2017
Project2_Blackjack
System Libraries:
iostream, string,
fstream, cstdlib,
ctime, iomanip
No Global Constants
Function Prototypes:
void dealCard(string card[], int&
randCard, string suit[], int&
randSuit);
void prntCard(string card[], int&
randCard, string suit[], int&
randSuit);
int calc(int& randCard, int& value);
A
A
Main
Random Seed:
strand(time(0))
Variable Declaration:
int value, playerTotal,
runTotal, dealerTotal,
dealrunTotal, randCard,
randSuit, choice;
char answer;
ifstream infile;
ofstream outfile;
bool exitMenu = true;
string welcome;
Variable Declaration:
string suit[4]={"of Hearts","of
Diamonds",
"of Spades","of Clubs"};
string card[14]=
{"Ace","2","3","4","5","6"
,"7","8","9","10","Ace",
"Jack","Queen","King"}
B
B
Output:
Game Menu
Input:
1
C
Input:
2
Output:
Game Rule
Input:
Any Key
Exit Main


C
Open Input File:
input.dat
Output:
Message
Close Input File:
input.dat
Function:
void dealCard(string card[], int& randCard, string suit[], int& randSuit);
void prntCard(string card[], int& randCard, string suit[], int& randSuit);
Value of Ace
Value = 1 ?
True
Value = 11
D
D
Calculate Player's Running Total:
playerTotal = runTotal + value;
runTotal = playerTotal;
Function:
void dealCard(string card[], int& randCard, string suit[], int& randSuit);
void prntCard(string card[], int& randCard, string suit[], int& randSuit);
calc(randCard, value);
Value of Ace for Additional Cards
runTotal>10 & value = 11 ?
True
Value = 1
runTotal<=10 & value =1 ?
True
Value = 11
Calculate Player's Score Total:
playerTotal = runTotal + value
Output:
Player's total
E
E
runTotal>10 & value = 11 ?
runTotal<=10 & value =1 ?
Function (Dealer):
void dealCard(string card[], int& randCard, string suit[], int& randSuit);
void prntCard(string card[], int& randCard, string suit[], int& randSuit);
calc(randCard, value);
Value of Ace
dealrunTotal>10 & value = 11 ?
True
Value = 1
dealrunTotal<=10 & value = 11 ?
True
Value = 11
Calculate Dealer's Running Total:
dealerTotal = dealrunTotal + value
dealrunTotal = dealerTotal
F


Function Definition:
Card Dealing Function
dealCard
void dealCard(string card[], int& randCard, string suit[], int& rand Suit)
Function Definition:
Card Dealing Function
Exit dealCard
Function Definition:
Function to Print Card
prntCard
void prntCard(string card[], int& randCard, string suit[], int& rand Suit)
Output:
card[randCard]
Output:
suit[randSuit]
Exit prntCard



Cross Reference for Project 2

Where in Code

Chapter	Section	Topic	Line number
2	2	cout	Used throughout entire project
	3	libraries	iostream, string, fstream, cstdlib, ctime, iomanip
	4	variables/literals	Lines 28-39
	5	Identifiers	Break, string, if, while, char, cin, etc….
	6	Integers	Follows my variables

	7	Characters	Lines 33
	8	Strings	Lines 34, 35, 39
	9	Floats No Doubles	
	10	Bools	Lines 38
	11	Sizeof *****	Achieved
	12	Variables 7 characters or less	Keep this consistent throughout project
	13	Scope ***** No Global Variables	
	14	Arithmetic operators	+
	15	Comments 20%+	Used throughout, better displayed in pseudo code
	16	Named Constants	
	17	Programming Style ***** Emulate	Achieved
3	1	cin	Lines 49, 101, 122, 142, 161
	2	Math Expression	Lines 78, 94, 116, 136, 156, 175, 199, 214
	3	Mixing data types ****	Demonstrated through ratio usage
	4	Overflow/Underflow ****	
	5	Type Casting	Line 43-48, 258-268
	6	Multiple assignment *****	Line 102, 123, 143, 162, 224
	7	Formatting output	Line
	8	Strings	Line 34, 35, 39, 277, 285
	9	Math Library	
	10	Hand tracing ******	Achieved
4	1	Relational Operators	Line
	2	if	Linem58, 74, 98, 102, 119, 123, 139, 143, 159, 162, 220,
	4	If-else	Line 184,
	5	Nesting	Most of the project is different nested statements. Primarily demonstrated in the menu function.

	6	If-else-if	Line 86-93, 110-115, 130-135, 150-155, 169-174, 193-198, 208-213, 224-252,
	7	Flags *****	
	8	Logical operators	
	11	Validating user input	Line
	13	Conditional Operator	Line
	14	Switch	Lines 51
5	1	Increment/Decrement	Line
	2	While	Line 63,
	5	Do-while	Line 41-273, 201-216,
	6	For loop	Line 327-330, 333-344
	11	Files input/output both	Line 57-67, 219-253
	12	No breaks in loops ******	
6	3	Function Prototypes	Line 277-283, 285-290, 292-324
	5	Passing by value	Line 324
	8	Returning values from functions	Line 279, 280
	9	Returning a boolean ******	
	10	No Global Variables Allowed	Achieved
		Only Global Constants	
		Meaning Conversions,Physical Constants only	
	11	Static Loca	
	12	Default arguments	Line 34, 35
	13	Reference Parameters	Line 324
	14	Overloading functions	
	15	Exit function *******	
7	4	Array Initialization	Line 334
	6	Processing Arrays	Line 335, 337, 339, 341, 343

	7	Parallel Arrays	
	8	Arrays as function arguments	Line 279-282
	9	2-D Arrays	Line 326-350
	12	STL Vector	
8	1	Linear and Binary Search	
	3	Bubble and Selection Sort	Line 277-283
	5	Search/Sorting Vectors ******	
****** Not	equired	show	

Pseudo Code

```
/*
  File:   main.cpp
  Author: Dokyoung Shin
  Created on June 2, 2017, 6:52 PM
  Purpose: Project 2 Black Jack Game.
 */

//System Libraries
// Needed to use Input and Output library
// Needed to use strings
// Needed to use stream Input and Output files
// Needed to use random function
// Needed to use time for random and program
```

```
// Needed to use control the Output format

//Global Constants
//Function Prototypes

//Execution Starts Here
//Main
    //Random Seed
    //Variable Declaration
    //Introduction Menu
    //Solve the problem chosen
    //Initialize Player and Dealer Running Totals
    //Open File input.dat
    //Function
    //Value of Ace
    //End of Value of Ace
    //Value of Ace for Additional Cards
    //End of Ace Re-Adjustment

                //Player can get additional card if total score is less than 21
                // First Player's Additional Card
                    //Players Additional Cards
                    //Value of Ace for Additional Cards
                     //End of Ace Re-Adjustment
                        //Second Additional Card
                                //Players Additional Card
                                 //Value of Ace for Additional Cards
                                  //End of Ace Re-Adjustment
```

```
                    //Third Additional Card
                      //Players Additional Card
                       //Value of Ace for Additional Cards
                        //End of Ace Re-Adjustment
                    //Fourth Additional Card
                      //Players Additional Card
                       //Value of Ace for Additional Cards
                        //End of Ace Re-Adjustment

            //Dealer's Card
                  //Value of Ace for Additional Cards
                   //End of Ace Re-Adjustment
                  //Value of Ace for Additional Cards
                   //End of Ace Re-Adjustment
            //Dealer can get additional card if total score is less than 16
            //Outcome - Open File output.dat to write result
             //Exit Stage Right

                     //Explains the Rule of BlackJack Game
             //Exit Stage Right
   //Randomly Selects Card and Suit
//Prints Card
   //Outputs One Card
//Function Calculates the Value of each card
   //Assigns Value to Card Dealt
//2D Array
```

Program Code

```
/*
  File:   main.cpp
  Author: Dokyoung Shin
  Created on June 2, 2017, 6:52 PM
  Purpose: Project 2 Black Jack Game.
 */

//System Libraries
#include <iostream> // Needed to use Input and Output library
#include <string> // Needed to use strings
#include <fstream> // Needed to use stream Input and Output files
#include <cstdlib> // Needed to use random function
#include <ctime> // Needed to use time for random and program
#include <iomanip> // Needed to use control the Output format
using namespace std;

//Global Constants

//Function Prototypes
void dealCard(string card[], int& randCard, string suit[], int& randSuit);
void prntCard(string card[], int& randCard, string suit[], int& randSuit);
int calc(int& randCard, int& value);

//Execution Starts Here
int main(int argc, char** argv) {
    //Random Seed
    srand(time(0));
```

```
//Variable Declaration
int value, playerTotal, runTotal;
int dealerTotal, dealrunTotal;
int randCard, randSuit;
int choice;
char answer;
string suit[4]={"of Hearts","of Diamonds","of Spades","of Clubs"};
string card[14]={"Ace","2","3","4","5","6","7","8","9","10","Ace","Jack","Queen","King"};
ifstream infile;
ofstream outfile;
bool exitMenu = true;
string welcome;

do{
   //Introduction Menu
   cout << "\n**********Game Menu**********\n";
   cout << "Enter 1 for Play BlackJack Game.\n";
   cout << "Enter 2 for See the Rule of Game.\n";
   cout << "Enter any key for Close the Game.\n";
   cout << "**********************************\n";
   cout << "\n Enter your choice: ";
   cin >> choice;//Input your choice
   //Solve the problem chosen
   switch(choice){
      case 1:
//Initialize Player and Dealer Running Totals
dealrunTotal = 0;
runTotal = 0;
```

```
//Open File input.dat
                infile.open("input.dat");
                   if(infile.fail())
                    {
                      cout << "Input file failed to open.\n";
                      exit(1);
                   }
                while (infile >> welcome)
                {
                   cout << welcome << " ";
                }
                infile.close();
                cout << endl << endl;
//Function
dealCard(card,randCard,suit,randSuit);
prntCard(card,randCard,suit,randSuit);
calc(randCard, value);
//Value of Ace
if(value == 1)
{
   value = 11;
}//End of Value of Ace
   playerTotal = runTotal + value;
   runTotal = playerTotal;
cout << " | ";
dealCard(card,randCard,suit,randSuit);
prntCard(card,randCard,suit,randSuit);
calc(randCard, value);
```

```
//Value of Ace for Additional Cards
if(runTotal>10&&value==11)
{
   value=1;
}
else if(runTotal<=10&&value==1)
{
   value=11;
}    //End of Ace Re-Adjustment
   playerTotal = runTotal + value;
   cout << "\n\nYour score is " << playerTotal << "." << endl;
                //Player can get additional card if total score is less than 21
                // First Player's Additional Card
                if(playerTotal<21)
                {
                 cout << "Would you like another card? [Y/N]\n";
                 cin >> answer;
                   if (answer =='Y'||answer =='y'){
                      //Players Additional Cards
                      cout << "\n";
                      cout << "You have been dealt a ";
                      dealCard(card,randCard,suit,randSuit);
                      prntCard(card,randCard,suit,randSuit);
                      calc(randCard, value);
                      //Value of Ace for Additional Cards
                       if(playerTotal>10&&value==11){
                          value=1;
```

```
 }
 else if(playerTotal<=10&&value==1){
    value=11;
 }//End of Ace Re-Adjustment
playerTotal = playerTotal + value;
cout << "\nYour new score is " << playerTotal << "." << endl;
    //Second Additional Card
    if (playerTotal<21)
    {
        cout << "Would you like another card? [Y/N]\n";
        cin >> answer;
        if (answer =='Y'||answer =='y'){
            //Players Additional Card
             cout << "\nYou have been dealt a ";
             dealCard(card,randCard,suit,randSuit);
             prntCard(card,randCard,suit,randSuit);
             calc(randCard, value);
             //Value of Ace for Additional Cards
              if(playerTotal>10&&value==11){
                 value=1;
              }
              else if(playerTotal<=10&&value==1){
                 value=11;
              }//End of Ace Re-Adjustment
             playerTotal = playerTotal + value;
    cout << "\nYour new score is " << playerTotal << "." << endl;
    //Third Additional Card
    if (playerTotal<21)
```

```
{
  cout << "Would you like another card? [Y/N]\n";
  cin >> answer;
  if (answer =='Y'||answer =='y'){
   //Players Additional Card
    cout << "\nYou have been dealt a ";
    dealCard(card,randCard,suit,randSuit);
    prntCard(card,randCard,suit,randSuit);
    calc(randCard, value);
    //Value of Ace for Additional Cards
     if(playerTotal>10&&value==11){
        value=1;
     }else if(playerTotal<=10&&value==1)
     {
        value=11;
     }//End of Ace Re-Adjustment
    playerTotal = playerTotal + value;
cout << "\nYour new score is " << playerTotal << "." << endl;
//Fourth Additional Card
if(playerTotal<21){
    cout << "Would you like another card? [Y/N]\n";
    cin >> answer;
  if (answer =='Y'||answer =='y'){
   //Players Additional Card
    cout<<"\nYou have been dealt a ";
    dealCard(card,randCard,suit,randSuit);
    prntCard(card,randCard,suit,randSuit);
    calc(randCard, value);
```

```
                //Value of Ace for Additional Cards
                 if(playerTotal>10&&value==11){
                    value=1;
                 }
                 else if(playerTotal<=10&&value==1){
                    value=11;
                 }//End of Ace Re-Adjustment
                playerTotal = playerTotal + value;
            cout << "\nYour new score is " << playerTotal << "." << endl;
              }
             }
           }
          }
           }
         }
      }
    else;
}
//Dealer's Card
cout << "\nThe house has been dealt the following cards: ";
cout << "\n";
        dealCard(card,randCard,suit,randSuit);
        prntCard(card,randCard,suit,randSuit);
        calc(randCard, value);
        //Value of Ace for Additional Cards
         if(dealerTotal>10&&value==11){
            value=1;
         }
```

```
        else if(dealerTotal<=10&&value==1){
           value=11;
        }//End of Ace Re-Adjustment
dealerTotal = dealrunTotal + value;
dealrunTotal = dealerTotal;
do
{
cout << " | ";
        dealCard(card,randCard,suit,randSuit);
        prntCard(card,randCard,suit,randSuit);
        calc(randCard, value);
        //Value of Ace for Additional Cards
         if(dealerTotal>10&&value==11){
           value=1;
         }
         else if(dealerTotal<=10&&value==1){
           value=11;
         }//End of Ace Re-Adjustment
dealerTotal = dealerTotal + value;
//Dealer can get additional card if total score is less than 16
}while(dealerTotal<16);
cout << "\nThe Dealer's score is " << dealerTotal << "." << endl;
//Outcome - Open File output.dat to write result
  outfile.open("output.dat", ios::app);
     if(outfile.fail()){
        cout << "Input file failed to open.\n";
        exit(1);
     }
```

```
if(playerTotal ==21||(playerTotal>dealerTotal && playerTotal<21)){
    cout << "\n";
    cout << "***Congratulations! You are the winner!***";
    outfile << "Win" << " " << endl;
    cout << "\n";
}
else if(dealerTotal>21 && playerTotal<=21) {
    cout << "\n";
    cout << "***Congratulations! You are the winner!***";
    outfile << "Win" << " " << endl;
    cout << "\n";
}
else if(playerTotal>21){
    cout << "\n";
    cout << "***Bust!***";
    outfile << "Loss" << " " << endl;
    cout << "\n";
}
else if(playerTotal<dealerTotal && dealerTotal<=21){
    cout << "\n";
    cout << "***Dealer Wins***";
    outfile << "Loss" << " " << endl;
    cout << "\n";
}else if(playerTotal==dealerTotal){
    cout << "\n";
    cout << "***Stand-Off/Draw, Play Again***";
    outfile << "Tie" << " " <<endl;
    cout << "\n";
```

```
                }
                 outfile.close();
                //Exit Stage Right
                      break;
                      case 2:
                         //Explains the Rule of BlackJack Game
                      cout << "**********BlackJack Game Rule**********\n";
                      cout << "Rule 1. The goal of Blackjack game is trying to make as close as 21 score."
                           "Moreover, the way to win the game is to beat the dealer by getting a higher score than "
                           "the dealer without going over 21 with any additional cards.\n";
                      cout << "Rule 2. All number cards are worth a number of points equal to their numerical value.";
                      cout << "Rule 3. In general, Ace is worth either 1 point or 11 points depends on the player's "
                           "decision. However, in this program, if player's total score is less than 10 points, "
                           "then program will add 11 points, if not it will add 1 point.\n";
                      cout << "Rule 4. Picture cards are worth 10 points.";
                      cout << "Kings = 10 points,\n Queens = 10 points,\n Jacks = 10 points\n";
                      cout<< "**************************************************";
                      cout << endl;
                      ;break;
                      default: exitMenu=false;
                   }
                }while(exitMenu);
                //Exit Stage Right
   return 0;
```

```
}
void dealCard(string card[], int& randCard, string suit[], int& randSuit){
   //Randomly Selects Card and Suit
   randSuit=rand()%4;
   randCard=rand()%14;
   card[randCard];
   suit[randSuit];
}
//Prints Card
void prntCard(string card[], int& randCard, string suit[], int& randSuit){
   //Outputs One Card
   cout<<right<<setw(2)<<card[randCard];
   cout<<" ";
   cout<<left<<setw(2)<<suit[randSuit];
}
//Function Calculates the Value of each card
int calc(int& randCard, int& value){
   //Assigns Value to Card Dealt
    if(randCard==0){
      value=1;
   }else if(randCard==1){
      value=2;
   }else if(randCard==2){
      value=3;
   }else if(randCard==3){
      value=4;
   }else if(randCard==4){
      value=5;
```

```
    }else if(randCard==5){
        value=6;
    }else if(randCard==6){
        value=7;
    }else if(randCard==7){
        value=8;
    }else if(randCard==8){
        value=9;
    }else if(randCard==9){
        value=10;
    }else if(randCard==10){
        value=11;
    }else if(randCard==11){
        value=10;
    }else if(randCard==12){
        value=10;
    }else if(randCard==13){
        value=10;
    }
     return value;
}
//2D Array
 void dealCard(int card[], int n, int randSuit){
    for(int i=0;i<n;i++){
        card[i][0]=rand()%14+1;
        card[i][1]=suit(randSuit);
    }
}
```

```
void prntCard(int card[], int n, int randSuit){
    for (int i=0;i<n;i++){
       cout<<card[i][0];
       cout<<" of ";
        if(card[i][0]==0){
            cout<<"Hearts";
        }else if(card[i][0]==1){
            cout<<"Diamonds";
        }else if(card[i][0]==2){
            cout<<"Clubs";
        }else if(card[i][0]==3){
            cout<<"Spades";
        }
    }
}
int suit(int& randSuit){
    randSuit=rand()%4+1;
    return randSuit;
}
```