Project 2: BlackJack

Miguel Alva Patino

CIS - 5 - 40107

February 10, 2019

Introduction:

Title: BlackJack

Description:

The object of the game is to be dealt cards to must add up to 21 points. The points/value for cards 2-10 are equivalent to their face number. Moreover, King, Queen and Jack cards are worth 10 points and an Ace can be either 1 point or 11 points, based on what the user thinks is best for his/her hand total. You will be dealt two cards initially and be asked if user would like to "hit" to get closer to 21. If user goes over 21, user "busts"/ loses.

In the beginning of the game, there is a display of the randomized numbers that will be dealt and then ask the user to mentally order them from ascending order. Then there will be a menu that will allow the user to play or exit the game. If the user decides to play and forgets the value of the cards, there will be a friendly reminder of the values of the cards.

Summary:

Code Line Total: 412

Loops: 9

Functions: 7

Checklist:

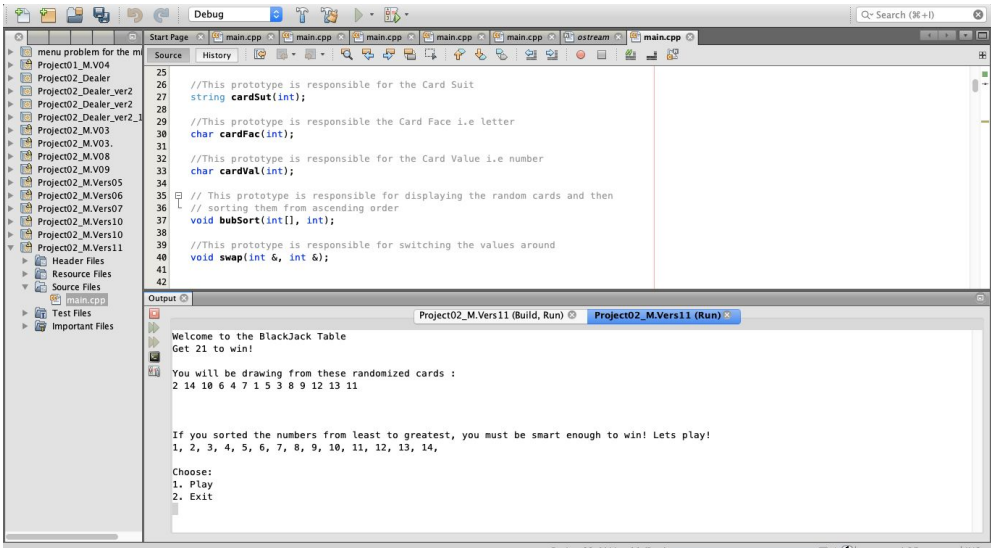
			Project 2: BlackJack Table of Contents		
Chapter	**Section**	**Topic**		**Line Occurance**	**Notes**
1 : Intro to programming	**5**	**Statements**			
2 : Intro to C++	**2**	**cout**		**38, 41,42,72,73,74,82,93,98,111, 112, 120**	
				121, 136,137,143,144,150,156,164,165,	
				175,182,187, 206, 211, 215, 221	
	3	**Libraries**		**9, 10, 11, 12, 13**	**iostream, iomanip, cstdlib, ctime ,cmath**
	4	**variables**			
	4	**literals**			
	6	***int* data type**		**32, 68, 127**	**int num1,num2,num3,num4, choice**
	7	***char* data type**		**33, 128**	**hit, letter**

3: Expressions and interativity	**1**	**cin**			**75,151**	**hit, choice**
	9	**Random Numbers**			**28,46,87,160**	**srand**
4: Making Decisions	**1**	**Relational Operators**			**46,50,52,55,57,60,62,77,87,92,94,97,99**	**+, =**
					88,102, 201 , 203 ,210 ,212 ,221 ,222	
	2	***if* Statement**			**49,54,59,70,91,96,101,109,153,162**	
					168,180	
	4	***if/else* Statement**			**209,205,200,214**	
	5	**nested *if* statement**			**134**	
	8	**The || operator**			**49,54,59,70,91,96,101,109,141,153,168**	**= =**
					180,200,209,227	
5: Loops and Files	**5**	**The *do-while* loop**			**130,227**	

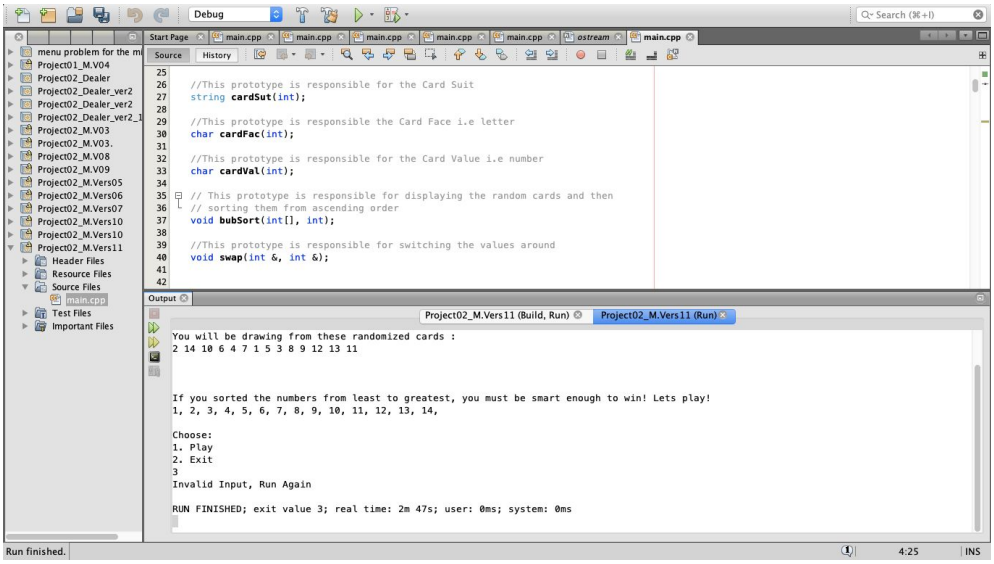
Chapter	Section	Topic			Line Occurrence	Notes
6 : Functions	2	Calling function: *void*			24,37,40,344,383,405	
	3	Function Prototype			24, 27,30,33,37,40	Void, void, void, char, char, string
	4	Sending Data into a function				
		Multiple arguments			37,40,45,84,383,405	
	7	*return*			115, 341, 360,369,380	return
	15	The *exit ()* function			121	exit(0)
7: Arrays and Vectors	1	Arrays hold multiple values			60, 62,360,369,395,397	
	2	Accessing Array Elements			74,91	
	4	Range-based *for* loop			74, 91, 390, 393	
	5	Processing Array Contents			366,369,395,397	
	6	Parallel Arrays				
	7	Arrays as Function Arguments			37,383	
8 : Searching and Sorting Arrays	1	Linear search				
	2	Binary Search				
	3	The Bubble Sort			37,84,383	
		Swapping array elements			40,397,405	
		The Selection Sort				

Screenshots: These screenshots are confirmation that my game ran using my personal NetBeans.

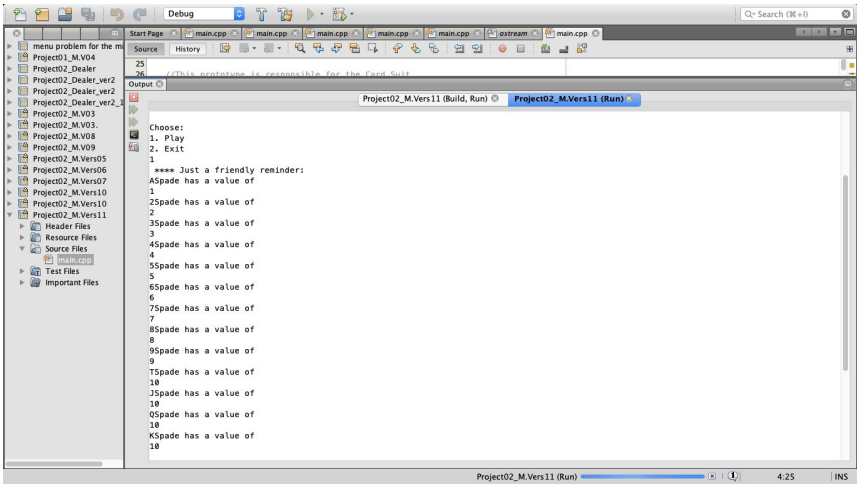
1.) Introduction to Project 2



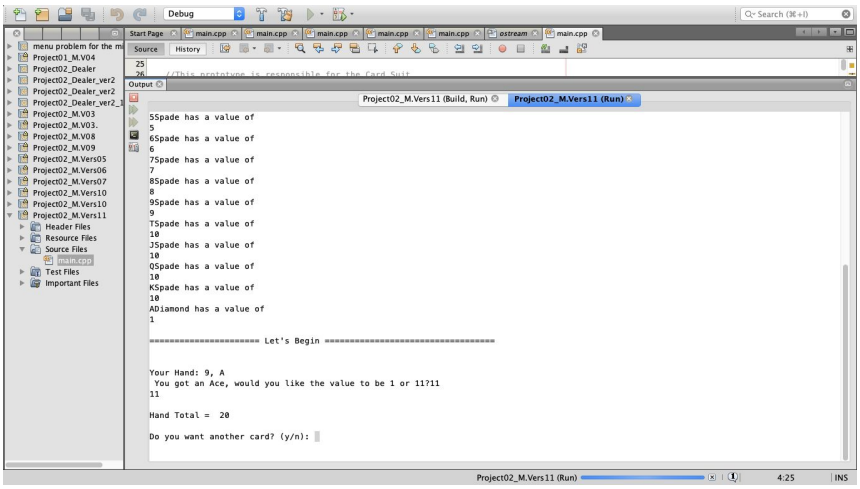
2.) Invalid Menu Input



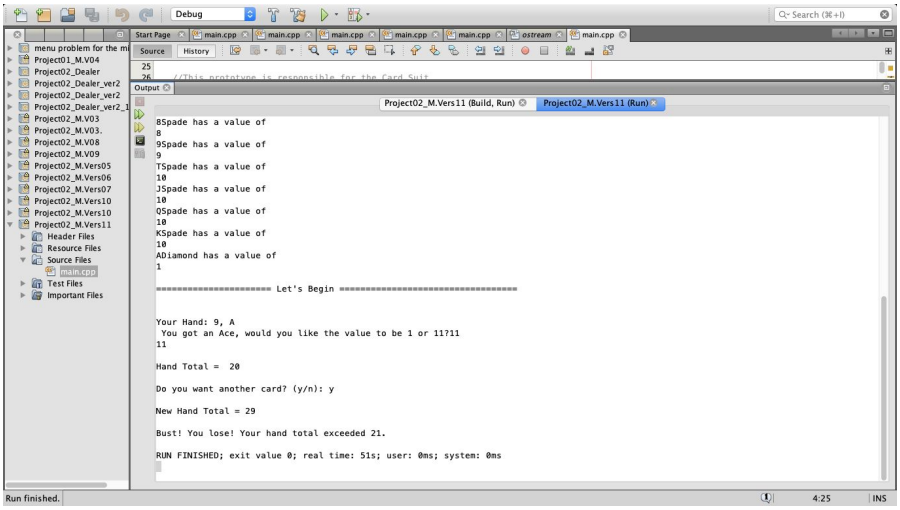
3.) The Value of the cards reminder



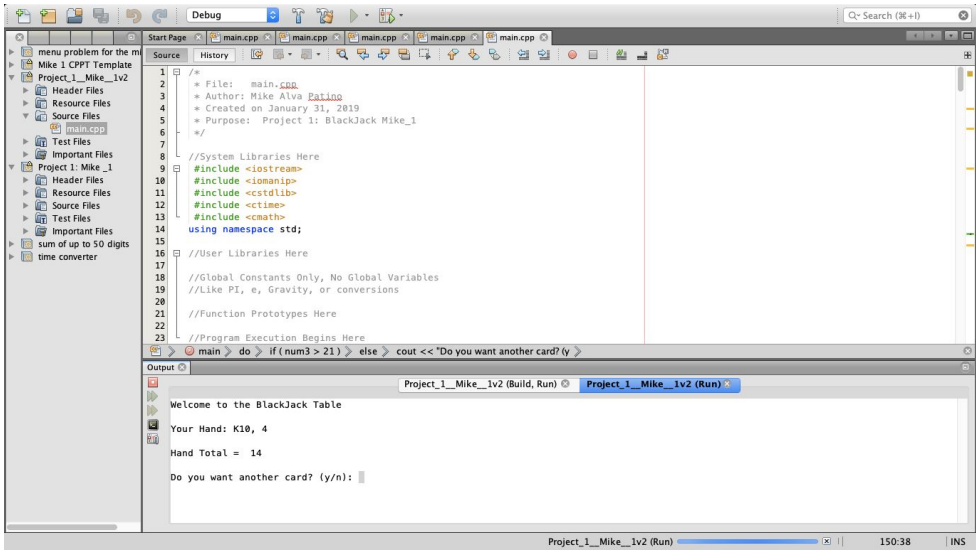
4.) Being dealt a card



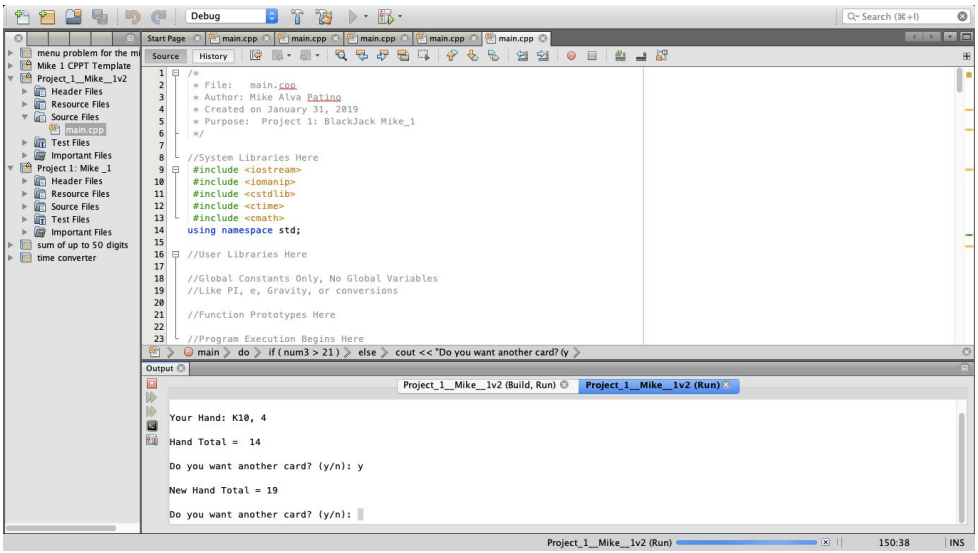
5.) Losing



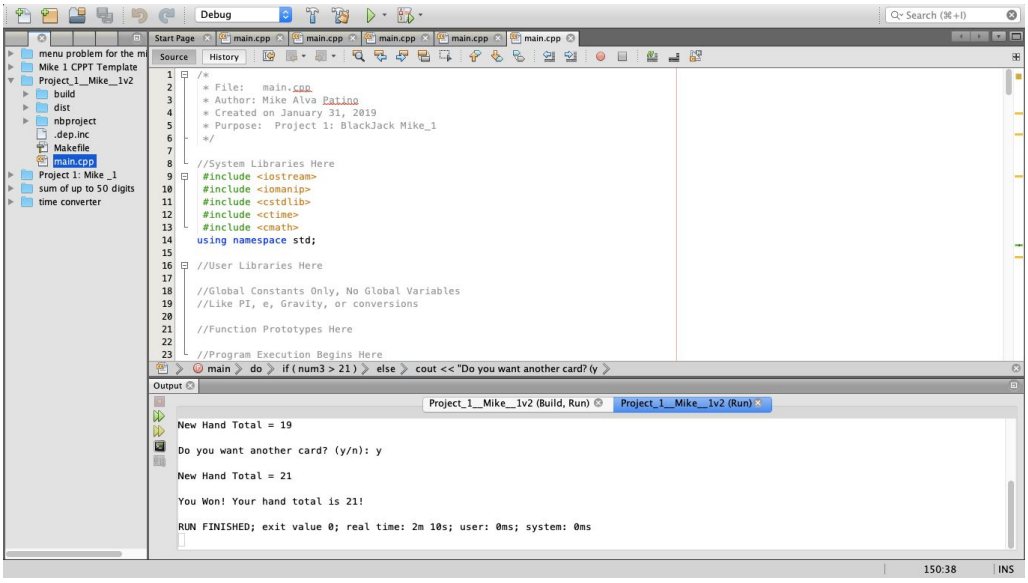
6.) Example of getting a King in first hand



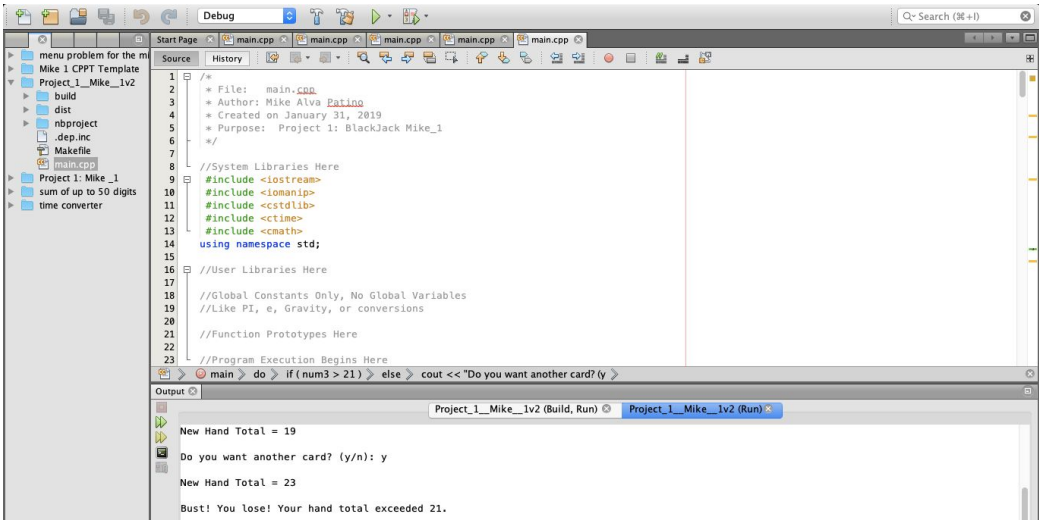
7.) Example of asking for another card



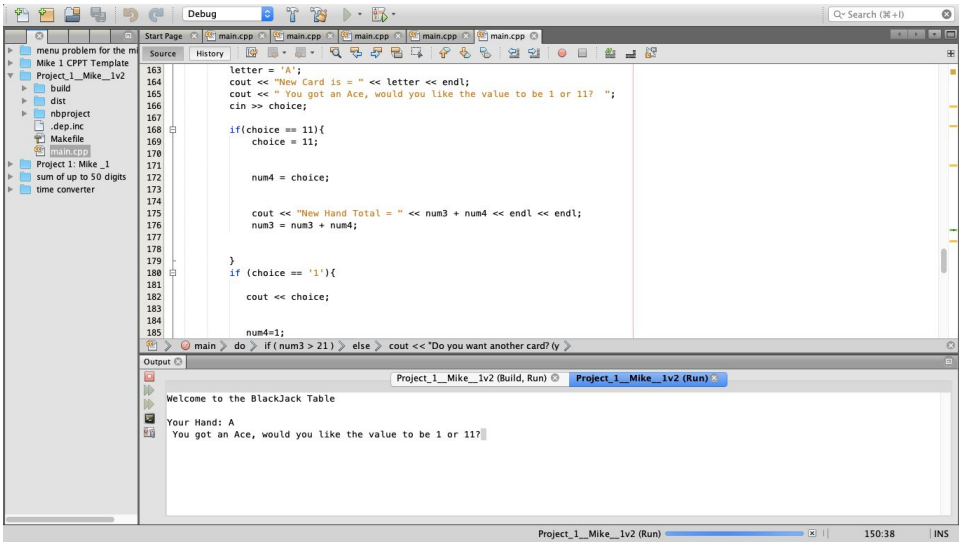
8.) Example of winning the game



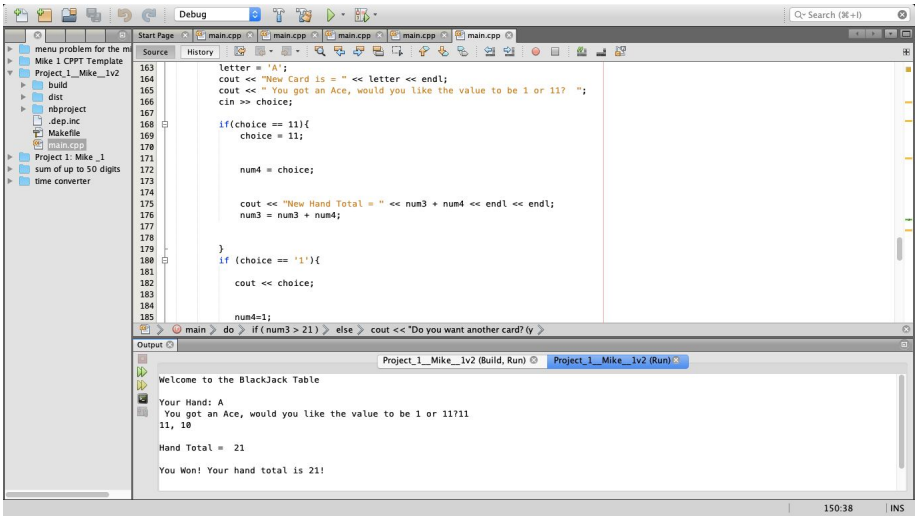
9.) Example of Losing the Game



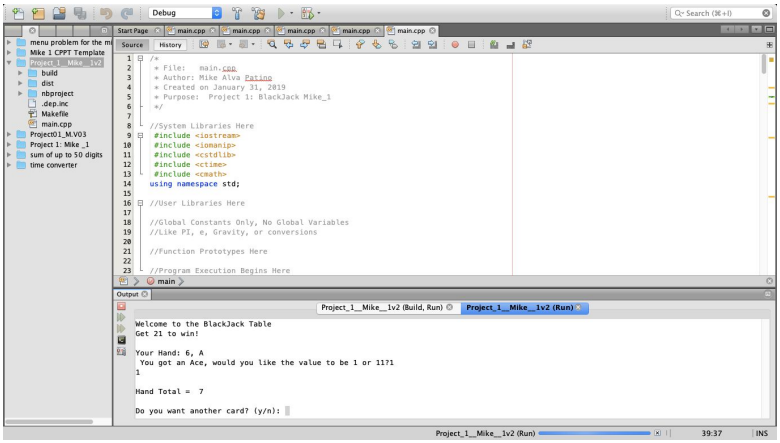
10.) Example of getting an Ace



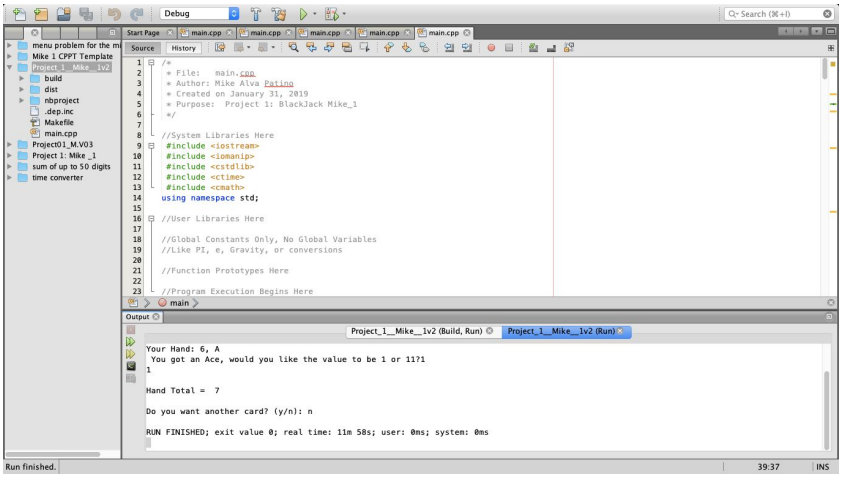
11.) Example of Using an Ace to win



12.) Example of an Ace being 1



13.) Not wanting another card



Flowchart:

In the folder as a PDF and XML file

Code:

```
/*
 * File:   main.cpp
 * Author: Mike Alva Patino
 * Created on January 31, 2019
 * Purpose:  Project 02: BlackJack
 */

//System Libraries Here
 #include <iostream>
 #include <iomanip>
 #include <cstdlib>
 #include <ctime>
 #include <cmath>
using namespace std;

//User Libraries Here

//Global Constants Only, No Global Variables
//Like PI, e, Gravity, or conversions

//Function Prototypes Here

//This Prototype is responsible for displaying the card face and value
```

```
void print(int);

//This prototype is responsible for the Card Suit
string cardSut(int);

//This prototype is responsible the Card Face i.e letter
char cardFac(int);

//This prototype is responsible for the Card Value i.e number
char cardVal(int);

// This prototype is responsible for displaying the random cards and then
// sorting them from ascending order
void bubSort(int[], int);

//This prototype is responsible for switching the values around
void swap(int &, int &);


//Program Execution Begins Here

int main(int argc, char** argv) {

    //Set the random number seed

     srand(static_cast<unsigned int>(time(0)));

    //Declare all Variables Here

    int num1, num2, num3;
    char letter;
    int NCARDS =14;
    int number;

  //For my Bubble Sort Function

    const int SIZE = 14;
```

```
    int val[SIZE] = { 2, 14, 10, 6, 4, 7, 1, 5, 3, 8, 9, 12, 13, 11};


//Initialize or input



   cout << "Welcome to the BlackJack Table " << endl;
   cout << "Get 21 to win!" << endl << endl;

   cout << "You will be drawing from these randomized cards : " << endl;

   for (auto element : val)
      cout << element << " ";

   cout << endl << endl << endl;


 //====================== Bubble Sort ============================//



   bubSort(val, SIZE);

   cout << endl;

   cout << "If you sorted the numbers from least to greatest, "
         "you must be smart enough to win! Lets play!" << endl;

   for (auto element : val){
      cout << element << ", " ;

   }

   cout << endl;
```

```
//Display the outputs


 //Sorted List
cout << endl;


cout << "Choose:" << endl;
cout << "1. Play" << endl;
cout << "2. Exit" << endl;
cin >> number;


while (number != 1 && number != 2){
    cout << "Invalid Input, Run Again" << endl;

    return number;


}

if (number == 2){
    exit (0);
}
if (number == 1){



cout << " **** Just a friendly reminder: " <<endl ;

//These are meant to be instructions or reminders of the values

print(NCARDS);

cout << endl;
```

```
  cout << "====================== Let's Begin
==================================" << endl;


  }
  cout << endl;

  cout << endl;
  cout << "Your Hand: ";



  //This will give user random numbers from [1-14]

   num1 = (rand() %13+1);

   //Card 1: Assigning King, Queen, Jack Values


   if (num1 == 11){
      letter = 'K';
      cout <<letter;
      num1 =10;
   }


      if (num1 == 12 ){
         letter = 'Q';
         cout <<letter;
         num1 =10;
      }


         if (num1 == 13){
            letter = 'J';
```

```
            cout << letter;
            num1 =10;
        }



//If Card 1 is an Ace


int choice;

if (num1 == 1 ){
    letter = 'A';
    cout <<letter;
    cout << endl;
    cout << fixed << setw(3) <<" You got an Ace, would you like the value to be 1 or 11?";
    cin >> choice;

    num1=choice;


}

        cout << num1 << ", ";


//Getting Card 2 From the Dealer

num2 = (rand() %13+1);

//Card 2: Assigning King, Queen, Jack Value

if (num2 == 11){
    letter = 'K';
    cout <<letter;
    num2 =10;
}
```

```
        if (num2 == 12){
            letter = 'Q';
            cout <<letter;
            num2 =10;
        }
            if (num2 == 13){
                letter = 'J';
                cout <<letter;
                num2 =10;
            }

    //Assigning Card 2 if it is an Ace

    if (num2 == 1){
        letter = 'A';
        cout <<letter;
        cout << endl;
        cout << fixed << setw(3) <<" You got an Ace, would you like the value to be 1 or 11?";

        cin >> choice;
        num2 = choice;
    }


        cout << num2 << endl << endl;
        cout << "Hand Total =  " << num1 + num2 << endl << endl;
        num3 = num1 + num2;


    //Declare New Variables for Card #3

    int num4;
    char hit;

do  {

    //This will decide whether or not the loop ends
```

```
if ( num3 > 21 ){

        cout << "Bust! You lose! Your hand total exceeded 21.";
        cout << endl;
                break;
}

else if (num3 == 21){

       cout << "You Won! Your hand total is 21!";
       cout << endl;
                break;
}

else {

cout << "Do you want another card? (y/n): ";
cin >> hit;

if (hit == 'y' || hit == 'Y' ){




cout << endl;

//This will give values from [1-14]

num4 = (rand() % 13+1);

 if (num4 == 1){
    letter = 'A';

    cout << " You got an Ace, would you like the value to be 1 or 11?  ";
    cin >> choice;
```

```
    if(choice == 11){
        choice = 11;


        num4 = choice;


        cout << "New Hand Total = " << num3 + num4 << endl << endl;
        num3 = num3 + num4;


    }
    if (choice == '1'){

       cout << choice;


       num4=1;

        cout << "New Hand Total = " << num3 + num4 << endl << endl;
        num3 = num3 + num4;

    }

    //Card 3: Assigning King, Queen, Jack Values



if (num4 == 12 ){
    letter = 'Q';
    cout <<letter;
    num4 =10;
}
else{
     cout << endl;
}
```

```
        if (num4 == 13){
            letter = 'J';
            cout << letter;
            num4 =10;
        }
            else{
             cout << endl;
            }

    }


       cout << "New Hand Total = " << num3 + num4 << endl << endl;
        num3 = num3 + num4;

        }
    }
}
while (hit == 'y' || hit == 'Y');

//Exit

return 0;
}

void print(int n){

        for(int i=0 ;i < n; i++ ){

            cout << cardFac(i) << cardSut(i)

            << " has a value of " <<static_cast<int> (cardVal(i)) <<endl;
            }
}
```

```
string cardSut(int i){

   string cardsut[] = {"Spade","Diamond" , "Clubs" , "Hearts"};

   return cardsut[i/13];

}

char cardFac(int i){

    char cardFac[] = {'A','2','3','4','5','6','7',
                      '8','9','T','J','Q','K'};

    return cardFac[i % 13];
}

char cardVal(int i){

    cout << endl;

    int imod13 = i%13+1;

        //This returns values 10 or lower

        return imod13 < 11 ? imod13:10;
}

void bubSort(int a[], int size){

    //This is the maximum element for sorting
    int maxElem;

    int indx;

    for (maxElem = size -1; maxElem > 0; maxElem-- ){
        for (indx =0 ; indx < maxElem; indx ++){
            if ( a[indx] > a[indx +1]){
```

```
            swap(a[indx], a[indx+1]);
         }
      }
   }

}

void swap(int &x, int &y){


   int temp = x;
   x = y;
   y = temp;

}
```

Pseudocode:

// Opening with file, name, date, and project name.
//Then utilize upstream, iomanip, cstdlib, ctime, cmath and using namespace std.
//Function prototypes are void print, string card suit, card face, card value bubble sort, and swap. They all have integer arguments.
// Now enter main function and immediately be given the srand static cast function with argument time 0.

//initializing variables num1, num2, num3 which are the user's "cards", char letter to represent the Jack, king, queen, and ace.
int NCARDS =14 to display the card values though a function and int number to use for the menu in the beginning of the program.

//constant int SIZE= 14 and the in val[SIZE]= to an array of mixed numbers from 1 to 14 which will be used in bubble sort.

//In the beginning, display “Welcome to the BlackJack Table Get 21 to win! You will be drawing from these randomized cards :” then for loop

//for (auto element : val)
  display the mixed values in the array

//call to bubble sort function

//Display “If you sorted the numbers from least to greatest, you must be smart enough to win! Lets play!” after the bubble sort

//for (auto element : val)
display the elements in ascending order

//allow the user to chose to play or exit by entering 1 or 2.

//while (number does not equal 1 and number does not equal 2)
display invalid input and run again with return number.

//if number is equal to 2
exit() function will exit the entire program

// if number is equal to 1 , display the called function print (NARDS) which will give a friendly reminder of the values of the cards

//num1 generates the random number generator from 1 though 14

//if num1 is equal to 11 then the value will be 10 and a K will display next to it
//if num1 is equal to 12 then the value will be 10 with a Q will be displayed next to it
//if num1 is equal to 13 then the value will be 10 and the letter J will display next to the value.

//int choice
//if num1 is equal to the 1 then the letter A will be displayed then the message alerting user of the ace and allow the user to input their choice
num1 will equal choice

//Display num1,

//num2 will be generated by the random number generator.

//if num2 is equal to 11 then the value will be 10 and a K will display next to it
//if num2 is equal to 12 then the value will be 10 with a Q will be displayed next to it
//if num2 is equal to 13 then the value will be 10 and the letter J will display next to the value.
//if num2 is equal to 1 the letter A will be displayed and a message alerting the ace will also be displayed
then enter the choice of either 1 or 11 to make that card the value
num2 will equal the choice value

//the hand total will be num 1 + num2
//num3 = num 1 + num2

//int num4, char hit;
//do { if num3 is greater than 21 then a message will indicate that the user busted and loses because they user exceeded 21 points
	//break;

//otherwise if (else if) num3 is equal to 21 then a display will state that the user won because the hand total is 21

//else does user want another card? (y/n)
//input hit

//if hit is equal to lowercase y or capital y
		then the unser will be given another card from the deck, num4 using a random number generator

//if num4 is equal to 1 then the letter A will be displayed and the user can choose the value of the ace

//if choice is equal to 11 then num4 equals 11
//if choice is equal to 1 then num4 will equal 1
//the New hand total is displayed along with num3+num4 which will accumulate the hand
//num3 = num3 +num4

//if num4 is equal to 11 then the value will be 10 and a K will display next to it

//if num4 is equal to 12 then the value will be 10 with a Q will be displayed next to it
//if num4 is equal to 13 then the value will be 10 and the letter J will display next to the value.
// while hit is equal to lower case y or capital Y

//return 0, which is the end of the main function

//in the function void print(int n)
//for the int i equals 0, and i is less than n, then display the cardFac(i) function and the cardSut (i) function as well. Then static cast the cardVal(i) function.

//next function is the string card suit (int i). This fills an array with character face values and returns the card face i%13

//the next function is the card value (int i) which returns values 10 or lower

//The next function void bubble sort (int a[], int size) is the function that will sort my filled randomized array.
//int maxElem, indx will fin the max element in the array and sort the numbers from ascending order.
//for the max element is equal to size-1 , and max element is greater than 0,max element - -
//only if the array [indx] is greater than array [indx +1]
//then utilize the swap function (array[indx], array[indx+1])

//The last function is the void swap (int &x, int &y) which will swap an int temp equals x, as x equals y and y equals temp.