```
/*****************************************************************************************************************
 * Program Name:        Highest Card Wins! v0.0019.1 (highcard.exe)
 * Course:              CIS-61, C++ Language Programming
 * Instructor:          C. Polen
 * Project:             Assignment 3
 * Created Date:        October 10th, 2010
 * Due Date:            October 15th, 2010
 * Created By:          Chad Philip Johnson
 * Purpose:             Play a really basic and boring game of War
 * Editor/IDE:          Notepad++
 * Resoluton:           1024x768
 * Compiler:            MinGW C++
 * Acknowledgements:    None
*****************************************************************************************************************
****/

#include <iostream>
#include <conio.h>  //for getch()
#include <cstdlib>  //for rand() and srand() functions
#include <ctime>
#include <string> //for type string
using namespace std;

struct card //declare structure for card player information
    {
    int     intCardNumber;  //for card number or kind
    int     intCardSuit;    //for card suit
    int     intTotalScore;  //combined score of card number/kind and suit
    string  strCardNumber;  //string to interpret card number or kind based on numerical value
    string  strCardSuit;    //string to interpret card suit based on numerical value
    };

void fncDrawCards();    //declare functions (see descriptions below)
void fncNameCards(card&);

card player01, player02;    //player 01 and 02 independant card values (player 02 is the computer)

/*****************************************************************************************************************
 * Function Name:       main()
 * Parameters:          None
 * Return Value:        int
 * Purpose:             Main program
*****************************************************************************************************************
****/

int main()
    {

```

```
47      srand( time(NULL) ); //Initiate random seed
48
49      char chrPrompt; //declare variables
50
51      cout << endl << endl << "Welcome to the the games Highest Card Wins! (v0.0019.1)" << endl << endl; //Description of the game
        and rules
52      cout << "The rules (very challenging):" << endl;
53      cout << "1)  You will draw one card." << endl;
54      cout << "2)  Your oppenent will draw one card." << endl;
55      cout << "3)  Whoever has the highest card wins." << endl << endl;
56      cout << "...and you can play as many times as you want!" << endl << endl;
57
58      cout << "Press any key to continue...." << endl << endl;
59      getch(); //Pause to display the results until user presses any button to continue
60
61      //plays game at least once
62      do {
63          fncDrawCards(); //assign player 01 and player 02 card values
64
65          cout << "You drew the " << player01.strCardNumber << " of " << player01.strCardSuit << "." << endl; //communicate player
            01 card
66          cout << "Press any key to continue...." << endl << endl;
67          getch(); //Pause to display the results until user presses any button to continue
68
69          cout << "Your cunning opponent drew the " << player02.strCardNumber << " of " << player02.strCardSuit << "." << endl;
            //communicate player 02 card
70          cout << "Press any key to continue...." << endl << endl;
71          getch(); //Pause to display the results until user presses any button to continue
72
73          if ( player01.intTotalScore < player02.intTotalScore )  //compare player 01 and player 02 total card score
74              cout << "Sorry, you lose!" << endl << endl; //lose when player 02 total score is greater
75          else
76              cout << "Congratulations, you win!" << endl << endl; //win when player 02 total score is lower
77
78          cout << "Would you like to play again? (y/n) "; //Prompt user if he/she would like to play again
79          cin >> chrPrompt;   //prompt for user input
80          cout << endl << endl << endl << endl;
81
82          } while ( chrPrompt != 'n' && chrPrompt != 'N' );   //continue playing so long as the player does not input 'n' or 'N'
83
84      return 0;   //end program
85
86      }
87
88
89  /*************************************************************************************************************************
    *****
90   * Function Name:       fncDrawCards()
91   * Parameters:          None
92   * Return Value:        None
```

```
 * Purpose:             Create random numbers and assign values to player's card and total
*********************************************************************************************************/

void fncDrawCards()
    {

    //assign random card values (number or kind and suit) for player 01
    player01.intCardNumber = (rand() % 13) + 2; //Add 2 to the random number generator (first value is 0; also ace equals 14, not 1)
    player01.intCardSuit = (rand() % 4); //card suit

    //assign random card values (number or kind and suit) for player 02
    player02.intCardNumber = (rand() % 13) + 2; //Add 2 to the random number generator (first value is 0; also ace equals 14, not 1)
    player02.intCardSuit = (rand() % 4); //card suit

    //continue to reassign new random values to player 02 until the total is not the same as the total for player 01
    while ( player01.intCardNumber == player02.intCardNumber && player01.intCardSuit == player02.intCardSuit ) {
        player02.intCardNumber = (rand() % 13) + 2; //Add 2 to the random number generator (first value is 0; also ace equals 14, not 1)
        player02.intCardSuit = (rand() % 4); //card suit
        }

    //calculate score totals
    player01.intTotalScore = player01.intCardNumber + player01.intCardSuit;
    player02.intTotalScore = player02.intCardNumber + player02.intCardSuit;

    //interpret card number or kind and suit integer values into english for player 01
    fncNameCards(player01);
    //interpret card number or kind and suit integer values into english for player 02
    fncNameCards(player02);

    }


/*********************************************************************************************************************
 * Function Name:       fncNameCards()
 * Parameters:          card (struct by reference)
 * Return Value:        None
 * Purpose:             Interpret card integer values to appropriate english string values
*********************************************************************************************************/

void fncNameCards(card& cardTemp)   //create new reference card structure for passed values (player 01 or player 02)
    {

    switch(cardTemp.intCardNumber)  //interpret card number or kind
        {
```

```
            case 2:
                cardTemp.strCardNumber = "2"; break;
            case 3:
                cardTemp.strCardNumber = "3"; break;
            case 4:
                cardTemp.strCardNumber = "4"; break;
            case 5:
                cardTemp.strCardNumber = "5"; break;
            case 6:
                cardTemp.strCardNumber = "6"; break;
            case 7:
                cardTemp.strCardNumber = "7"; break;
            case 8:
                cardTemp.strCardNumber = "8"; break;
            case 9:
                cardTemp.strCardNumber = "9"; break;
            case 10:
                cardTemp.strCardNumber = "10"; break;
            case 11:
                cardTemp.strCardNumber = "jack"; break;
            case 12:
                cardTemp.strCardNumber = "queen"; break;
            case 13:
                cardTemp.strCardNumber = "king"; break;
            case 14:
                cardTemp.strCardNumber = "ace"; break;
            }

        switch(cardTemp.intCardSuit)    //interpret card suit
            {
            case 0:
                cardTemp.strCardSuit = "clubs"; break;
            case 1:
                cardTemp.strCardSuit = "diamonds"; break;
            case 2:
                cardTemp.strCardSuit = "hearts"; break;
            case 3:
                cardTemp.strCardSuit = "spades"; break;
            }

        }

```