

C++ Review

Write a complete C++ program, including at least one comment for the main program and one for each function, as follows. Your program will emulate some aspects of a card game based on the game of Hearts. It's okay if you don't know how to play Hearts—all the information you need is described below.

In the program, a playing card is represented as a number between 0 and 51. Each card has a *suit* and a *face*. The suit can be either: “diamonds”, “clubs”, “hearts” or “spades”. The face can be either the numbers 2 through 10, or “jack”, “queen”, “king” or “ace”. In our representation, we can convert a value between 0 and 51 to its “suit” and “face” as follows:

```
int suit = (int)( card / 13 );
int face = card % 13;
```

Your job is to fill in the missing pieces of the program:

1. the function showCard()
2. the function showHand()
3. the function countPoints()
4. the function change3cards()
5. the main() function

```
//----- SOURCE CODE BEGINS HERE -----
```

```
// include C and C++ headers
#include <stdlib.h>
#include <time.h>
#include <iostream>
using namespace std;

// define constants for suits
const int DIAMONDS = 0;
const int CLUBS    = 1;
const int HEARTS    = 2;
const int SPADES    = 3;

// define constants for face characters, such that:
// 0 is '2', 1 is '3', ... , 8 is ten ('T'), 9 is jack ('J'),
// 10 is queen ('Q'), 11 is king ('K'), 12 is ace ('A')
//      0123456789012
const string FACES = "23456789TJQKA";
```

```

/**
 * dealHand()
 * this function emulates dealing "numcards" from a deck.
 * it stores the cards in the "cards" array.
 * it ensures that there are no duplicates in the "cards" array.
 */
void dealHand( int numcards, int cards[] ) {
    bool picked[52]; // flags indicating if a card value has been picked or not
    int newcard; // value of new card
    srand( time( NULL ) ); // initialize random number seed
    for ( int i=0; i<52; i++ ) { // initialize "picked" flags
        picked[i] = false;
    }
    for ( int i=0; i<numcards; i++ ) { // pick "numcards" unique cards
        newcard = rand() % 52; // find a card that hasn't been picked yet
        while ( picked[newcard] ) {
            newcard = rand() % 52;
        }
        cards[i] = newcard; // select the unpicked card
        picked[newcard] = true; // set flag indicating the card has been picked
        cout << "value of card[" << i << "] = " << cards[i] << endl;
    }
} // end of dealHand()

/**
 * showCard()
 * this function displays the face character and suit name of the
 * "card" argument. for example, if "card" is 0, then this function
 * will output "2 of diamonds".
 */
void showCard( int card ) {
    // YOU NEED TO WRITE THE CODE FOR THIS FUNCTION
} // end of showCard()

/**
 * showHand()
 * this function displays the face character and suit name of each entry in the
 * "cards" array. "numcards" is the number of entries in this array.
 * HINT: call the function "showCard()", which you have defined above,
 * to display the face character and suit name of individual entries.
 */
void showHand( int numcards, int cards[] ) {
    // YOU NEED TO WRITE THE CODE FOR THIS FUNCTION
} // end of showHand()

```

```

/**
 * countPoints()
 * this function will count and return the number of points in the hand
 * represented by the "cards" argument. "numcards" is the number of entries in
 * this array.
 * points are allocated as follows: the queen of spades (value 49) is worth -26
 * points. all hearts are worth 1 point. cards of all other suits are worth 0.
 */
int countPoints( int numcards, int cards[] ) {
    // YOU NEED TO WRITE THE CODE FOR THIS FUNCTION
} // end of countPoints()

/**
 * change3cards()
 * This function will select 3 entries in the "cards" argument that should be
 * changed. "numcards" is the number of entries in this array. The "change"
 * argument is an array of size 3.
 * First, the function should look for the queen of spades (value=49) in the
 * "cards" array. If it finds it, then it should put its value in the "change"
 * array, and replace its entry in the "cards" array with -1.
 * Second, the function should find the 2 or 3 largest values in the "cards"
 * array, place these values in the "change" array and place -1 in each entry in
 * the "cards" array that is selected.
 * Note that we say "2 or 3 values" above because if the queen of spades is
 * found, then you only need to find the 2 largest values; otherwise you need to
 * find the 3 largest values.
 */
void change3cards( int numcards, int cards[], int change[] ) {
    // YOU NEED TO WRITE THE CODE FOR THIS FUNCTION
} // end of change3cards()

/**
 * main()
 * This is the main function of the program. Inside, you should write code to:
 * 1. declare an array of 13 integers that will store an array of cards that
 *    represents your "hand" in the game
 * 2. call the function dealHand() to initialize your array ("hand")
 * 3. call the function showHand() to display the contents of your hand
 * 4. call countPoints() to calculate the point value of the cards in your hand,
 *    and then display that value
 * 5. call change3cards() to select 3 cards to change from your hand,
 *    and then display those 3 cards
 */
int main() {
    // YOU NEED TO WRITE THE CODE FOR THIS FUNCTION
} // end of main()

```