COMSC 200

Summer 2024

Programming Assignment 6

Worth 12.5 points (1.25% of your grade)

DUE: Monday, 7/1/24 by 11:59 P.M. on Canvas

Late Pass Deadline: Thursday, 7/4/24 by 11:59 P.M. on Canvas

Your solution should for this assignment should consist of **six (6)** files:

Card.h (class specification file)

Card.cpp (class implementation file)

DeckOfCards.h (class specification file)

DeckOfCards.cpp (class implementation file)

200_assign6.cpp (application program)

Sample_runs.pdf (sample runs)

Please continue to use the same **naming convention** as before, where each filename should contain both your first name and your last name. If your first name is “James” and your last name is “Smith”, then your header file should be named James_Smith_Card.h. The other five files should use the same naming convention.

Comments – worth 1.25 points (10%) of your programming assignment grade:

Your program should have at least **ten (10)** different detailed comments explaining the different parts of your program. Each individual comment should be, at a minimum, a sentence explaining a particular part of your code. You should make each comment as detailed as necessary to fully explain your code. You should also number each of your comments (i.e., comment 1, comment 2, etc.).

Sample Runs – worth 1.25 points (10%) of your programming assignment grade:

You should submit screenshots of at least **five (5)** different sample runs of your program. You should also number each of your sample runs (i.e., sample run 1, sample run 2, etc.). **NOTE: Your sample runs should be different from my sample runs shown in this write-up for the programming assignment, and each of your five sample runs should show the 52 cards in a different order.** Each sample run should show the 52 cards in the deck. For example:

```
King of Spades      Jack of Spades      Seven of Spades     Five of Hearts
Queen of Diamonds   Queen of Hearts     Three of Hearts     Deuce of Spades
Three of Diamonds   Four of Clubs       Five of Spades      Five of Diamonds
Ten of Diamonds     Ace of Hearts       Four of Hearts      Six of Spades
Six of Clubs        Ten of Clubs        Nine of Clubs       King of Diamonds
Deuce of Clubs      Jack of Diamonds    Eight of Clubs      Eight of Spades
Seven of Clubs      King of Clubs       Nine of Spades      Queen of Clubs
Queen of Spades     Eight of Hearts     Eight of Diamonds   Jack of Clubs
Jack of Hearts      Five of Clubs       Three of Clubs      Deuce of Diamonds
Nine of Diamonds    Seven of Hearts     Four of Diamonds    Ten of Spades
Ace of Diamonds     Ten of Hearts       Deuce of Hearts     Ace of Spades
Three of Spades     Six of Hearts       Ace of Clubs        Four of Spades
King of Hearts      Seven of Diamonds   Nine of Hearts      Six of Diamonds
Press any key to continue
```

For your sixth programming assignment you will be writing a program to shuffle and deal a deck of cards. The program should consist of class Card, class DeckOfCards, and an application program.

Class Card should provide:

a. Data members face and suit of type int. These are **index numbers** for the faces and suits arrays described in step c below.
b. A constructor that receives two ints representing the face and suit and uses them to initialize the data members.
c. Two static arrays of strings representing the faces and suits. The faces array stores the strings “Ace”, “Deuce”, “Three”, “Four”, “Five”, “Six”, “Seven”, “Eight”, “Nine”, “Ten”, “Jack”, “Queen”, and “King”. The suits array stores the strings “Hearts”, “Diamonds”, “Clubs”, and “Spades”.
d. A toString function that returns the Card as a string in the form ***“face** of **suit”.*** For example, “Ace of Hearts”. You can use the + operator to concatenate strings.

Class DeckOfCards should contain:

a. A vector of Cards named deck (**of type Card** – the Card class) to store the Cards. There are 52 cards in a deck, which is the number of faces (13) times the number of suits (4).*
b. An integer currentCard representing the next card to deal. This is used as an **index numbe**r for the deck vector from step a.
c. A default constructor that initializes the Cards in the deck. You can initialize the faces and suits in the order shown above in step c of the Card class.**
d. A shuffle function that shuffles the Cards in the deck. The shuffle algorithm should iterate through the array of Cards. For each Card, randomly select another Card in the deck and swap the two Cards.
e. A dealCard function that returns the next Card object from the deck vector.
f. A moreCards function that returns a bool value indicating whether there are more Cards to deal. This function will return false if all 52 cards have been dealt, otherwise it returns true.

*Note that this is **composition**: a deck of cards **has** cards, so the DeckOfCards class has a vector of **type Card** as one of its private member variables

This means that each index in the deck vector stores **an entire Card object,** and each Card object in turn consists of a face (type int) and a suit (type int). For example:

deck[0] = Card(0, 0)
deck[1] = Card(0, 1)
….
deck[51] = Card(12, 3)

** The faces array stores the strings “Ace”, “Deuce”, “Three”, “Four”, “Five”, “Six”, “Seven”, “Eight”, “Nine”, “Ten”, “Jack”, “Queen”, and “King”. The suits array stores the strings “Hearts”, “Diamonds”, “Clubs”, and “Spades”. You can initialize the arrays following this order.

The application program should create a DeckOfCards object, shuffle the cards, and then deal the 52 cards. Each time you run the program you should see all 52 cards, but they should be displayed in a **different order** each time the program is run (as shown below).

Sample run 1:

```
King of Spades     Jack of Spades     Seven of Spades    Five of Hearts
Queen of Diamonds  Queen of Hearts    Three of Hearts    Deuce of Spades
Three of Diamonds  Four of Clubs      Five of Spades     Five of Diamonds
Ten of Diamonds    Ace of Hearts      Four of Hearts     Six of Spades
Six of Clubs       Ten of Clubs       Nine of Clubs      King of Diamonds
Deuce of Clubs     Jack of Diamonds   Eight of Clubs     Eight of Spades
Seven of Clubs     King of Clubs      Nine of Spades     Queen of Clubs
Queen of Spades    Eight of Hearts    Eight of Diamonds  Jack of Clubs
Jack of Hearts     Five of Clubs      Three of Clubs     Deuce of Diamonds
Nine of Diamonds   Seven of Hearts    Four of Diamonds   Ten of Spades
Ace of Diamonds    Ten of Hearts      Deuce of Hearts    Ace of Spades
Three of Spades    Six of Hearts      Ace of Clubs       Four of Spades
King of Hearts     Seven of Diamonds  Nine of Hearts     Six of Diamonds
```

Sample run 2:

```
Seven of Diamonds  Ten of Hearts      Six of Hearts      Queen of Diamonds
King of Hearts     Three of Clubs     Deuce of Spades    Five of Clubs
Ten of Diamonds    Five of Spades     Ace of Diamonds    Nine of Spades
Nine of Diamonds   Ace of Hearts      Eight of Diamonds  Eight of Hearts
Jack of Hearts     Deuce of Clubs     Four of Hearts     Nine of Clubs
Eight of Spades    Jack of Diamonds   Five of Hearts     Four of Diamonds
Four of Clubs      Three of Spades    Six of Diamonds    Queen of Spades
Jack of Spades     Deuce of Diamonds  King of Diamonds   Jack of Clubs
Four of Spades     Seven of Hearts    Queen of Hearts    Seven of Spades
Eight of Clubs     King of Clubs      Ten of Clubs       Six of Spades
Seven of Clubs     Deuce of Hearts    King of Spades     Nine of Hearts
Ten of Spades      Three of Hearts    Ace of Clubs       Three of Diamonds
Six of Clubs       Ace of Spades      Five of Diamonds   Queen of Clubs
Press any key to continue . . .
```

Sample run 3:

```
Nine of Diamonds   Queen of Clubs     Seven of Diamonds  Ace of Diamonds
King of Hearts     Nine of Clubs      Three of Diamonds  King of Clubs
Ten of Hearts      Eight of Clubs     Seven of Clubs     Eight of Diamonds
Six of Diamonds    Five of Hearts     King of Spades     Three of Clubs
Queen of Spades    Three of Spades    Seven of Spades    Ten of Clubs
Eight of Spades    Ace of Hearts      Four of Hearts     Jack of Spades
Seven of Hearts    Six of Clubs       Four of Spades     Jack of Diamonds
Ace of Clubs       Nine of Hearts     Jack of Clubs      Five of Spades
Deuce of Diamonds  Six of Hearts      Nine of Spades     Eight of Hearts
Four of Diamonds   Deuce of Spades    Six of Spades      Five of Diamonds
Queen of Diamonds  Jack of Hearts     Ten of Diamonds    Queen of Hearts
King of Diamonds   Five of Clubs      Deuce of Hearts    Ace of Spades
Deuce of Clubs     Three of Hearts    Ten of Spades      Four of Clubs
Press any key to continue . . .
```

Sample run 4:

```
Queen of Hearts    Four of Spades     King of Clubs      Queen of Spades
Ace of Spades      Five of Hearts     Nine of Spades     King of Spades
Three of Hearts    Six of Spades      Ten of Spades      Four of Diamonds
King of Diamonds   Four of Hearts     Deuce of Clubs     Deuce of Hearts
Seven of Diamonds  Four of Clubs      Five of Spades     Ten of Diamonds
Ten of Clubs       Seven of Clubs     Eight of Diamonds  Nine of Hearts
Jack of Clubs      Six of Clubs       Seven of Spades    Three of Diamonds
Ace of Clubs       Queen of Diamonds  Queen of Clubs     Ten of Hearts
Six of Hearts      Eight of Spades    Deuce of Spades    Deuce of Diamonds
Six of Diamonds    Three of Spades    Jack of Spades     King of Hearts
Eight of Hearts    Five of Clubs      Eight of Clubs     Ace of Hearts
Five of Diamonds   Three of Clubs     Nine of Diamonds   Seven of Hearts
Jack of Hearts     Jack of Diamonds   Ace of Diamonds    Nine of Clubs
Press any key to continue . . .
```