Programming Assignment #3

Due Oct 15 by 3pm **Points** 100 **Submitting** a file upload
Available Oct 3 at 8pm - Dec 5 at 12pm 2 months

This assignment was locked Dec 5 at 12pm.

Assignment #3: Object-Oriented Programming

For this assignment, create a robust C# console application that implements the following functionality:

Simulate **a standard deck of 52 playing cards** **(https://en.wikipedia.org/wiki/Standard_52-card_deck)** using Object Oriented Programming techniques that we discussed in class.

1.) Create a deck of cards.
2.) Display your deck of cards.
3.) Randomly shuffle your deck of cards.
4.) Display your shuffled deck of cards.
5.) "**Cut** **(https://en.wikipedia.org/wiki/Cut_(cards))** " the shuffled deck of cards using a randomly chosen cut point.
6.) Display your cut deck of cards.
7.) Deal 4 hands of 5 cards each to 4 players.
8.) Display each player's hand as well as the cards remaining in the deck.

Sample Output:

```
Assignment 3:
==================================

New deck:
[[Ace of Clubs], [Two of Clubs], [Three of Clubs], [Four of Clubs], [Five of Clubs], [Six of Clubs], [Seven of Cl
ubs], [Eight of Clubs], [Nine of Clubs], [Ten of Clubs], [Jack of Clubs], [Queen of Clubs], [King of Clubs], [Ace
of Diamonds], [Two of Diamonds], [Three of Diamonds], [Four of Diamonds], [Five of Diamonds], [Six of Diamonds],
[Seven of Diamonds], [Eight of Diamonds], [Nine of Diamonds], [Ten of Diamonds], [Jack of Diamonds], [Queen of Di
amonds], [King of Diamonds], [Ace of Hearts], [Two of Hearts], [Three of Hearts], [Four of Hearts], [Five of Hear
ts], [Six of Hearts], [Seven of Hearts], [Eight of Hearts], [Nine of Hearts], [Ten of Hearts], [Jack of Hearts],
[Queen of Hearts], [King of Hearts], [Ace of Spades], [Two of Spades], [Three of Spades], [Four of Spades], [Five
of Spades], [Six of Spades], [Seven of Spades], [Eight of Spades], [Nine of Spades], [Ten of Spades], [Jack of Sp
ades], [Queen of Spades], [King of Spades]]
 52 cards in deck.

Shuffled deck:
[[Four of Diamonds], [Three of Clubs], [Five of Clubs], [King of Spades], [Ace of Clubs], [King of Hearts], [Jack
of Clubs], [Two of Hearts], [Jack of Spades], [Ten of Diamonds], [Ten of Hearts], [Three of Hearts], [Five of Hea
rts], [King of Diamonds], [Two of Clubs], [Queen of Diamonds], [Ace of Diamonds], [Ace of Hearts], [Jack of Heart
s], [Three of Spades], [Ace of Spades], [Seven of Diamonds], [Queen of Hearts], [Three of Diamonds], [Eight of He
```

```
arts], [Six of Hearts], [Ten of Clubs], [Nine of Spades], [Seven of Spades], [Six of Spades], [Nine of Diamonds],
[Four of Hearts], [Ten of Spades], [Five of Diamonds], [Eight of Clubs], [Eight of Spades], [Jack of Diamonds],
[King of Clubs], [Nine of Clubs], [Two of Spades], [Six of Clubs], [Seven of Clubs], [Four of Clubs], [Six of Dia
monds], [Four of Spades], [Queen of Clubs], [Eight of Diamonds], [Queen of Spades], [Two of Diamonds], [Five of S
pades], [Nine of Hearts], [Seven of Hearts]]
 52 cards in deck.

Cut deck:
[[Two of Diamonds], [Five of Spades], [Nine of Hearts], [Seven of Hearts], [Four of Diamonds], [Three of Clubs],
[Five of Clubs], [King of Spades], [Ace of Clubs], [King of Hearts], [Jack of Clubs], [Two of Hearts], [Jack of S
pades], [Ten of Diamonds], [Ten of Hearts], [Three of Hearts], [Five of Hearts], [King of Diamonds], [Two of Club
s], [Queen of Diamonds], [Ace of Diamonds], [Ace of Hearts], [Jack of Hearts], [Three of Spades], [Ace of Spade
s], [Seven of Diamonds], [Queen of Hearts], [Three of Diamonds], [Eight of Hearts], [Six of Hearts], [Ten of Club
s], [Nine of Spades], [Seven of Spades], [Six of Spades], [Nine of Diamonds], [Four of Hearts], [Ten of Spades],
[Five of Diamonds], [Eight of Clubs], [Eight of Spades], [Jack of Diamonds], [King of Clubs], [Nine of Clubs], [T
wo of Spades], [Six of Clubs], [Seven of Clubs], [Four of Clubs], [Six of Diamonds], [Four of Spades], [Queen of
Clubs], [Eight of Diamonds], [Queen of Spades]]
 52 cards in deck.

Dealt hands:
[[Queen of Spades], [Six of Diamonds], [Two of Spades], [Eight of Spades], [Four of Hearts]]

[[Eight of Diamonds], [Four of Clubs], [Nine of Clubs], [Eight of Clubs], [Nine of Diamonds]]

[[Queen of Clubs], [Seven of Clubs], [King of Clubs], [Five of Diamonds], [Six of Spades]]

[[Four of Spades], [Six of Clubs], [Jack of Diamonds], [Ten of Spades], [Seven of Spades]]

Remaing cards in deck:
[[Two of Diamonds], [Five of Spades], [Nine of Hearts], [Seven of Hearts], [Four of Diamonds], [Three of Clubs],
[Five of Clubs], [King of Spades], [Ace of Clubs], [King of Hearts], [Jack of Clubs], [Two of Hearts], [Jack of S
pades], [Ten of Diamonds], [Ten of Hearts], [Three of Hearts], [Five of Hearts], [King of Diamonds], [Two of Club
s], [Queen of Diamonds], [Ace of Diamonds], [Ace of Hearts], [Jack of Hearts], [Three of Spades], [Ace of Spade
s], [Seven of Diamonds], [Queen of Hearts], [Three of Diamonds], [Eight of Hearts], [Six of Hearts], [Ten of Club
s], [Nine of Spades]]
 32 cards in deck.
```

Important Notes:

- Before starting, be sure to review the information in **PREPARING FOR PROGRAMMING ASSIGNMENT #3**
- You must use Noun/Verb analysis to better define your Classes, Properties and Methods before starting to write your program
- You will need at least 2 enums and 3 classes (in addition to the Application class) for this assignment.
- **Console.WriteLine() calls should only appear in your Main() method.** Override ToString() to generate output for Console.WriteLine() to display.
- Be sure to read, understand and follow **[0070] Our C# Coding Standards**
- Do not use any C# language features that have not be covered in the reading or discussed in class - that still includes the Collection classes! (I PROMISE we will get to them next week!). Standard arrays

should do fine for now.

- Always choose descriptive variable, parameter, and method names. Always strive to make your code self-documenting.
- When you are finished, please ZIP up the Visual Studio Solution directory for your program and upload it using the link on this page.

My Grading Guideline:

40% Does the program compile and run?
30% Does the program generate correct results?
10% Does the program exhibit good OO characteristics?
10% Are your POJO classes "bulletproof"?
10% Is your code readable/maintainable/extendable and correctly formatted?