Card Game



Problem Statement

* Given a deck of cards, develop a console application.
* User playing the game should have three options
  + Play a card
    - When user plays a card – the application must throw the card i.e. the card at the at the top of a shuffled deck is shown.
    - The card played should be in a random order
    - The card once played would not appear again while playing
  + Shuffle the deck
    - User can shuffle the cards in hand (cards which are not played yet)
  + Restart the game
    - Mid way while playing, user may decide to play again.
    - Restarting a game will have all the 52 cards again in hand.

Evaluation Criteria

* Code Clarity, ease of understanding
* OOPs fundamentals, how to extend your solution to other applications
* Code commenting
* Code maintainability, how easy is it to maintain the code, or easy is it to extend the changes to the application
* Exception handling
* Class, Function, Variable naming conventions

Appendix

* A deck has 52 playing cards
* There are 4 suits each (Clubs, Hearts, Spades, Diamond)
* Each suit has 13 cards starting from Ace (A) till King (K)
  + A, 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K = 13 cards

Assessing your code

**Comment**:

The application was built on the following understanding

1. Game play is card on card and the game gets over when one player puts same card on top of the card in the game stack.
2. Both user and computer hold a deck of card each (52 cards each) so as to compare the Card on value and suite.

The main data structure used in the application building is **STACK** to take the benefit of Last In First Out feature.

1. What additional changes would you want to bring to your code, which you could not do in the given time frame.
2. Following shall be the changes I would like to bring to the code
3. Currently application is a 1 player game, i.e. player against computer. To be able to make it as a multi-player game.
4. More comments in the application and test cases
5. Improving shuffle mechanism. On reading through internet found that there are 52! Different decks possibility and random can achieve only 2^32 different possibility. Suggestion to use RNGCryptoServiceProvider to be able to generate all possible decks.

Reason for not able to touch on the above additional changes were the time constraint. Its weekdays and office itself is hectic. The application was implemented and tested in 5 hours.

Also, above points are just improvements.

1. Do you think your code has following attributes? If NOT, then explain why?
   1. Code Clarity
   2. Follows OOPs fundamentals
   3. Code commenting
   4. Code maintainability
   5. Exception handling
   6. Proper naming conventions

Yes I think I have covered all attributes mentioned above.

On code clean-up I might be able to reduce line of code and may be improve the user interactivity.