

Assignment 2

Due Thursday, January 20th

Write a C# program to implement a Card and Deck class. The requirements are below. Here is a test program that you will use to run against your classes. Provide a copy of its output.

```
using System;

namespace Assignment2
{
    class Class1
    {
        [STAThread]
        static void Main(string[] args)
        {
            Deck d = new Deck();
            Console.WriteLine("Initial Deck is = {0}",d);
            Console.WriteLine("");

            // write some cards using the indexer
            Console.WriteLine("{0}",d[5]);
            Console.WriteLine("{0}",d[22]);
            Console.WriteLine("{0}",d[31]);
            Console.WriteLine("");

            // remove some cards
            d.remove( 5 );
            d.remove( 22 );
            d.remove( 31 );
            Console.WriteLine("with cards removed = {0}",d);
            Console.WriteLine("");

            // reset and shuffle
            d.reset();
            d.shuffle();
            Console.WriteLine("New shuffled deck = {0}",d);
            Console.WriteLine("");
        }
    }
}
```

Requirements:

1. Your Card class should have:
 - a. An enumeration for the suite of a card (Clubs, Diamonds, Hearts, Spades). Name this enumeration Suite.
 - b. Two protected (or private) fields. One of type int for the value of the card (Ace=1, two, ..., Jack=11, Queen = 12, and King = 13) and one of type Suite (see (a) above).

- c. Three properties as follows:
 - i. Property pip that sets and gets the value of a card. This should check for valid values. If not valid,, the value should be set to 0.
 - ii. Property suite that sets and gets the Suite of a card.
 - iii. Property pip10 that gets the value of a card using the value 10 for Jacks, Queens and Kings. No set is needed.
- d. Two constructors as follows:

```
public Deck() - creates the card Ace of Spades
public Deck(int pip, Suite s) - creates the card pip of s
```

2. The Deck class should have:

- a. Two protected or private fields: an array holding 52 Cards and an integer variable holding the number of cards left in the deck.
- b. A property to get the number of cards remaining.
- b. A default constructor that sets up the deck with 52 cards and sets the number of cards variable to 52.
- c. A number of methods:
 - i. A method to shuffle the deck
 - ii. A method to reset the deck to starting conditions
 - iii. A method to remove the ith card from the deck. This should check if i is valid.
- d. An indexer to give the ith card in the deck. This should check if i is valid. If not, return null.

3. For both classes, provide a member function that overrides the Object class's ToString method. The signature is:

```
public override string ToString()
```