Homework #1: Cards

Due: Tuesday, January 27 @ 12:30 PM

Submission:

Please turn in all files on Canvas before the deadline. You should compress your submission into a single file, do not submit a large number of individual files. If you know you are going to miss a deadline, contact the TA **before** the deadline. Canvas has been known to be quirky, so it is not advised to wait until 5 minutes before it is due to make your submission.

Please include a text file called “README” at the top level of your main project directory. Include the following:

- Your name
- Your email address
- How long this project took you to complete
- Any comments or notes for the grader

Overview:

This is not a group assignment. It is acceptable to consult with other class members, **but your code must be your own**.

You will be creating a simple iPhone application using concepts you have learned in the first several lectures. You will become familiar with the basics of Objective C syntax and with development in the Xcode IDE.

Specifications:

Part 1: Model

You will create four classes: Card, Deck, PlayingCard, and PlayingDeck.

Class Card will represent a generic card, e.g. a flash card, a playing card, a pokemon card, etc. Internally the value of the card will be represented by a string. For example, an ace of clubs from a deck of playing cards would be represented internally by the string “A♣”. There should be a property of class Card called “contents” that contains this representation.

Class Deck will represent a generic collection of cards. Deck will have methods to insert a new card at a given position, remove a card at a given position, and to shuffle the cards.

Class PlayingCard will be a subclass of Card. Playing cards all have a suit (♠,♣,♥, or ♦) and a rank (1, 2, 3, …, 11, 12, or 13). The suit and rank of a PlayingCard should be represented internally as properties. The setter for these properties should ensure that only valid choices are used. Override the getter for the “contents” property of Card so that it properly returns a valid representation of the card. A card of rank 1 (Ace) should be represented by 'A'. A card of rank 11 (Jack) should be represented by 'J'. A card of rank 12 (Queen) should be represented by 'Q'. A card of rank 13 (King) should be represented by 'K'.

Class PlayingDeck will be a subclass of Deck. When PlayingDeck is initialized it will contain all 54 cards.

Part 2: View

Change the background color of your app to something other than the default.

Choose an image for the back of your cards. The front face of your cards should be white. (If you really want to, feel free to make your card design fancy.)

Your app should display a single card in the middle of the screen. The card may start out face up or face down. On the back of the card is an image of your choosing. The front of the card should display the “contents” property. Render the text for ♠ and ♣ black, the text for ♥ and ♦ red.

Include the following somewhere on the display:

- The number of cards still in the deck
- How many cards have been turned over
- How many times the deck has been reshuffled

Part 3: Control

Everytime you touch the card it should flip over. If it was face up it becomes face down. If it was face down it becomes face up. Every time you turn the card over you should change it into a randomly chosen new card. Every time you choose a new card remove it from the deck. When there are no more cards remaining, add them back to the deck, re-shuffle, and continue.

Notes:

You may use any code presented in class. Please type the code yourself (as opposed to copy-paste) as it is a good learning experience.

Before you start this assignment, you will need to download and install Xcode using the App Store on Mac OSX (not the App Store on your iOS device, the App Store on your Mac). It is free.

Economy is valuable in coding: the easiest way to ensure a bug-free line of code is not to write the line of code at all. This assignment requires few lines of code so if you find yourself writing more than a handful of lines of code (over and above what was demonstrated in lecture), you are on the wrong track.