CS161: Introduction to Computer Science
Homework Assignment 9
Due: Friday 4/15 by 11:59pm

Sets of Cards

This assignment is a pair assignment for which you have a week and a half to work. Please find someone to work with and then *email me your names* so I can make sure everyone has a partner.

In this assignment, you will be fully implementing the `CardSet` class from the midterm exam. Recall that a `CardSet` represents a generic "set of cards". This set of cards could be a player's hand, or a deck of cards, or a discard pile, etc. Each of these examples is fundamentally just a set of cards.

The difference between this assignment and the exam question is that you will be using an array to implement your set of cards instead of an `ArrayList`. Practically, this means that all of the nice built-in behavior of an `ArrayList` – e.g., growing the array, inserting and removing from the middle of the array – you will now implement yourself.

This assignment has 3 classes: `Card`, `CardSet`, and `Controller`. I have provided you with my `Card` class to use. Most of your time will be spent implementing the `CardSet` class.

The `Controller` class is a chance for you to actually use the `CardSet` class that you wrote. Inside the `Controller` class, you will use your `CardSet` class to assemble a standard deck of cards.

Designing the `CardSet` class

Imagine that you have a set of cards. Take a moment and discuss with your partner the different actions you would do with this set of cards in real life. (Seriously, talk with your partner about this.)

When I sat down and thought about this question, here is what I came up with: add a card to a particular spot in the set, remove a card from a particular spot in the set, shuffle all the cards, and maybe show all of the cards. Each of these actions then becomes a method in the `CardSet` class[1].

Here is a list of the methods that your `CardSet` class should have:

1. `CardSet()` – A constructor that takes no input parameters
2. `void add(Card card, int index)` – A void method that adds a card to the set at the specified index. If there is already a card at that spot, then you should shift that element and any subsequent elements to the right.
3. `Card remove(int index)` – A method that removes the `Card` at the given index from the set. This method should shift any subsequent elements to the left to fill the gap. This method should also return back the card that was removed.
4. `void shuffle()` – A method that shuffles the cards in the set. To implement this method, you should loop through every card in the array and swap it with another randomly chosen card from the array. Use a `Random` object to find a card to swap with.

[1] Section 7.2 in the textbook tries to formalize this process of *program design* – i.e., determining what classes, objects, methods, and variables are needed to implement a program. The exercise of asking yourself, "How would this object be used in real life?" is one way of identifying the methods that should belong in a class.

5. `String toString()` – A method that returns a string representation of the set. Please put each card on a separate line.

In addition to these methods, there are a few considerations you need to keep in mind. Both the `add` and `remove` methods take an integer index. *You must check that this index is valid.* If it is not valid, then do not carry out the action. Please do not print an error message.

Second, when you add a card to the set, you must make sure there is enough room in the array itself. If not, increase the capacity of the set and then add the card.

The `Controller` class

When you spend a lot of time coding a class, you want to benefit from all your hard work by actually using it to do something! Ideally, if we had more time, you would use your `CardSet` class to program an actual card game. However, for this assignment, you're just going to create a standard 52 card deck.

Create a Java class named `Controller`. Inside the `Controller` class, add a `main` method that uses a `CardSet` to hold a standard 52 card deck. Please print out the deck of cards, shuffle it, and then print it again. Finally, please remove some cards from your deck to show that your `remove` methods works.

Style Guide

Before you submit your assignment, go through the checklist below and make sure your code conforms to the style guide.

Checklist

- ☐ All unused variables are deleted
- ☐ All instance variables are used in more than one method (if not, make them local)
- ☐ Javadoc comment for all classes
- ☐ All methods have Javadoc comments (except for the `main` method)
- ☐ All numbers have been replaced with constants (i.e. no magic numbers)
- ☐ Proper capitalization of variables, methods, and classes
- ☐ Use white space to separate different sections of your code

Read the "Style Guide" (under "Resources" on the course website) for more information.

Submitting your homework assignment

You should submit your `hw9` folder with the `Card`, `CardSet`, and `Controller` class inside. Please rename your folder with both of your names before you zip it.