Mr. Levin AP CS JAVA

4.1 Cards (Adapted from official AP Java Labs)

In this lab you will be using everything you have learned about object oriented design, and the Java programming language to implement a digital deck of cards. In the extension you will use this deck of cards to create a card game.

https://docs.oracle.com/javase/8/docs/api/java/util/ArrayList.html

Challenge 1: Card Class

Fields:

1. **String rank(name of the card)**
2. **String suit**
3. **int pointValue**

Methods:

1. **Constructor (String rank, String suit, int pointValue)**
2. **Getters for all three fields.**
3. **public boolean equals(Card otherCard) //check if this card equals another in value.**
4. **Public String toString() - return an appropriate string for this card**

Challenge 2: Deck Class

The Deck Class uses an ArrayList to store a deck of cards.

Read up on ArrayLists here: http://beginnersbook.com/2013/12/java-arraylist/

Fields:

1. ArrayList<Card> unDealt; //an arraylist of all the cards in the deck

2. ArrayList<Card> Dealt; //an arrayList we will use to hold cards we have already seen, so we can shuffle them back in.

Methods:

1. constructor — This constructor receives three arrays as parameters. The arrays contain
the ranks, suits, and point values for each card in the deck. The constructor creates an
ArrayList, and then creates the specified cards and adds them to the list.

For example, if ranks = {"A", "B", "C"}, suits = {"Giraffes", "Lions"},
and values = {2,1,6}, the constructor would create the following cards:
["A", "Giraffes", 2], ["B", "Giraffes", 1], ["C", "Giraffes", 6],
["A", "Lions", 2], ["B", "Lions", 1], ["C", "Lions", 6]
and would add each of them to cards.

2. isEmpty — This method should return true when the size of the deck is 0; false otherwise.

3. size — This method returns the number of cards in the deck that are left to be dealt.

4. deal — This method "deals" a card by removing a card from the deck and returning it, if there are any cards in the deck left to be dealt. It returns null if the deck is empty. The dealt card should be added to the Dealt list and returned.

5. shuffle – This method shuffles the cards from the Dealt list back into unDealt, and shuffles the unDealt list into a somewhat random order.

You will be using Math.random to create randomness. Implement the selection shuffle algorithm below.

The "efficient selection shuffle," works as follows:
For k = 51 down to 1,
− Generate a random integer r between 0 and k, inclusive;
− Exchange cards[k] and cards[r].
This has the same structure as selection sort:
For k = 51 down to 1,
− Find r, the position of the largest value among cards[0] through cards[k];
− Exchange cards[k] and cards[r].

<u>Challenge 3 DeckTester:</u>

Use the main method in the DeckTester Class to run tests while you are developing this lab. Make sure you test the full functionality.

1. Build a deck of cards.

2. Shuffle the cards.

3. Deal out all the cards (for our purposes you can just print them out as you deal them.)