Viewing a Deck of Cards

(designed to accompany the AP® Computer Science A Elevens Lab)

- Consider the following questions first:

1. In the code, what is a deck of cards?
2. How would you look through all the Cards in a Deck?
3. Remember: to view specific characteristics of a card you must … deckName.get(indexOf).characteristic
4. What code would need to be in a toString() method that returns the deck inventory? Consider the following:
   - Returning the deck inventory means that a rather large amount of String information is returned.
   - Giving the method the name toString() poses an issue, do you know how to deal with it?
   - Define a String variable named deckInventory that will story all of the deck information
   - Remember that a+=3 means to add 3 to a if a is an int? Well, if a="hi", then a+="bob" equals "hibob".
   - How could you use a for-loop to add each line of output without manually coding it?
   - Consider the following output that should occur when the deck toString() method is called

```
Deck contains 4 cards. They are ...
Card #1: A of DOG, value = 6
Card #2: B of PIG, value = 7
Card #3: C of COW, value = 8
Card #4: D of BAT, value = 9
```

- There are no instance variables to be declared in this lab as you will be adding a method to an already-created class:

- Below is the framework for the basic deck toString() method. How would you complete it?

```
public String toString(){
        //code that declares/initializes the String variable named deckInventory
        //code utilizing a for-loop to add each line of output (communicating each card in the deck)
        //code to return the entire deck inventory (which is now stored in the deckInventory variable)
 }
```

- Use the following code to begin your DeckTester Class (starts with only 4 cards to simplify a deck of cards)

```
String[] types = {"A","B","C","D"};                 // The 4 types of cards for this card game
String[] suits = {"DOG","PIG","COW","BAT"};         // The 4 pictures shown on the cards for this card game
int[] values = {6,7,8,9};                           // The 4 point-values for cards in this card game

Deck myDeck = new Deck(types,suits,values);         // Builds a deck of 4 cards for this card game
```

- Add code to the rest of the DeckTester Class that does the following…

1. Calls the Deck toString() method which then in turn prints the entire deck inventory.
2. Verify that the output matches the output that is shown above (in the first section of this lab above).

- SPECIAL NOTE: Currently the inventory IS NOT printed in the order that the cards will be dealt. In fact, it is printed in reverse order. To fix this issue, the for-loop header line can be changed to count backwards instead of forwards. Complete this task and verify that your DeckTester gives the following output:

```
Deck contains 4 cards. They are ...
Card #1: D of BAT, value = 9                    //Be sure the cards are printed in this order with this numbering
Card #2: C of COW, value = 8
Card #3: B of PIG, value = 7
Card #4: A of DOG, value = 6
```