CS 251
Program 02

Main topics: Multiple Classes
Declaring / Using Instance Variables
Driver Classes

Program Specification:

You are to write a Class `Deck` which emulates a full deck of playing cards. That is 4 suits (Clubs, Spades, Hearts, and Diamonds) and 13 ranks (Ace, 2, 3, 4, 5, 6, 7, 8, 9, Jack, Queen, King) in each suit. This of course makes for a total of 52 playing cards in the deck.

Mandatory Instance variable:

```
private boolean[] deck = new boolean[52];
```

Mandatory Instance and Class methods:

```
public void initDeck()
// set the values of deck to indicate that they are all
// pressent - not delt yet.

public boolean emptyDeck()
// returns wheather or not all the cards in the deck
// have already been delt.

public int dealCard()
// returns a card (an int in the range 0 to 51) at random
// that has not been delt since the deck was initialize
// via intDeck.  Also notes (in deck) that this card is
// no longer available.

public static String cardToString(int card)
// given a card (an int in the range 0 to 51) returns
// an appropriate String repressentation of this card
// based on a 1-1 and onto mapping of the set [0, 51]
// to the cards described above.
```

You are also to write a **Driver Class** `DeckDriver` to test your `Deck` class.

Mandatory Functionality:
Your driver class must minimally print all the cards in the deck in the random order that they are "dealt". Such as in Program 1.

Rules and Requirements:

- All access to the instance variable(s) in your deck classes' instance methods must be made via *this*.

Notes and Hint:

1. You should be able to re-use much of your methods code from Program 1 in writing your deck class.

2. You should be able to "re-write" your main method from Program 1 into your driver class with minimal modification / effort.

Lastly you are to write a second deck class `SmartDeck` which adds a second instance variable `cardsDealt` that at all times contains the number of cards dealt since that last call to `initDeck()`

Notes and Hint:

1. `cardsDealt` will need to be modified by `initDeck()`, and `dealCard()`, and will allow you to write `emptyDeck()` without the use of a loop.
2. Your `DeckDriver` class must also work identically whether "myDeck" is declared as `Deck` or `SmartDeck`.

Sample run(s):

```
Run 1: - with Deck class -
------------------------------------------------------------
Here is a shuffled deck ...

7S  KS  2H  6S  4C  2D  9D  9C
4H  7C  9H  3D  5H  5D  10S  2S
JH  AH  4S  KC  QC  AD  QD  7D
AS  KD  5C  7H  KH  3C  JC  2C
4D  8H  AC  5S  10C  JS  3H  9S
8D  10D  8S  6C  QH  8C  JD  3S
QS  6D  10H  6H


Run 2: - with SmartDeck class -
------------------------------------------------------------
Here is a shuffled deck ...

2D  10C  AD  6C  JC  JH  KS  4S
9C  9S  2S  AC  QS  3C  3H  8C
3S  QC  AS  4D  10S  2C  8S  6D
6S  9H  2H  5S  JD  KD  QH  10D
7H  QD  3D  6H  7D  8H  5D  4H
KH  AH  8D  7C  9D  7S  5C  5H
KC  JS  4C  10H
```

Submission:

1. Use your web browser to open:

   `https://uwm.edu`

2. Select [Current Students] from the top menu bar
3. Select [Canvas] from the drop down menu
4. Login to Canvas
5. Click on the COMPSCI 251 block

6. Click on Assignments
7. Click on Program 02 in the left center of the current window
8. Click the **Submit Assignment** button in the right top of the current window
9. Click the **Browse** button in the left center top of the current window
10. Use the *File Upload* pop-up window to find the file you wish to submit
11. Click on this file name in the right panel of the *File Upload* pop-up window
12. Click the **Open** button in the *File Upload* pop-up window
13. Click the **Add** button in the bottom right top of the *Submit a File* pop-up window
14. Click the **Submit Assignment** button in the left bottom of the current window