

# CS 251

## Program 04

---

Main topics: Writing Classes  
Declaring / Using Instance Variables  
Writing Instance Methods  
Accessors and Mutators  
Constructors  
public vs private  
static vs non-static

### 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 variables:

```
private Card[] deck = new Card[52];
private int cardsDealt;
```

Mandatory Instance methods:

```
public Deck() // constructor
// set each element of deck to a unique Card object,
// and sets cardsDealt to zero.

public int getCardsDealt() // accessor
// return the value of cardsDealt.

private void setCardsDealt(int cardsDealt) // mutator
// sets cardsDealt specified value (cardsDealt)

public boolean isEmptyDeck()
// returns wheather or not all the cards in deck
// have already been dealt (cardsDealt == 52).

public void collectAllCards()
// set cardsDealt to zero.

public Card dealCard()
// if emptyDeck() is false ...
// returns the card at location cardsDealt in deck,
// and increments cardsDealt by 1.
// else ...
// returns null

public void shuffleDeck()
// apply 100 random card swaps within deck

public void shuffleDeck(int swapCnt)
// apply swapCnt random card swaps within deck
```

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 Programs 1, 2, and 3.

Rules and Requirements:

- All access of the instance variable `cardsDealt`, by the other instance methods is made via its accessor and mutator.

Notes and Hint:

1. You will need to copy your `Card.java` file (from Program 3) into your working directory / project in order to compile your `deck` class.
2. You should be able to re-use much of your methods code from Programs 1, 2, and 3.
3. You should be able to "re-write" your driver class from Program 3 into your driver class with minimal modification / effort.

Sample run(s):

**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 251-401 block
6. Click on Assignments
7. Click on Program 04 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