# Add a Shuffle Method to Deck (AP)

**INTRODUCTION**

Shuffling a deck of card mixes up the order of the cards. After a few shuffles, the order of the cards ends up looking pretty random. The results of one particular shuffle is called a permutation. If you shuffle again, you have another order of cards, which means another permutation. With 52 cards in a deck, there are 52 factorial permutations. 52 factorial is 52*51*50*49*48 .... *1 so the number of permutations of a deck of cards is extremely large. There are 8.0658 X $10^{67}$ ways to uniquely order a deck of cards, permutations.

> **Materials**
>
> - Computer with BlueJ IDE

## Procedure

You implemented a `Deck` class in AP Activity 4.2.2. The `Deck` class should be complete except for the `shuffle` method. You also implemented a `DeckTester` class that you used to test your incomplete `Deck` class. In AP Activity 4.2.3, you implemented methods in the `Shuffler` class, which shuffled integers. Now, you will use what you learned about shuffling to implement the `Deck shuffle` method.

1.  Open and create a BlueJ project for *ElevensActivity4* with the source code from your teacher. The file `Deck.java` contains a correct solution from AP Activity 4.2.2.

2.  Complete the Deck class by implementing the shuffle method. Use the efficient selection shuffle algorithm from AP Activity 4.2.3.

> **NOTE**
>
> The `Deck` constructor creates the deck and then calls the `shuffle` method. The `shuffle` method also needs to reset the value of `size` to indicate that all of the cards can be dealt again.

**3** The `DeckTester` class provides a basic set of `Deck` tests. It is similar to the `DeckTester` class you might have written in AP Activity 4.2.2. Add additional code at the bottom of the `main` method to create a standard deck of 52 cards and test the `shuffle` method. You can use the Deck `toString` method to "see" the cards after every shuffle.

## CONCLUSION

1. How does the `Deck` constructor provided in this activity compare to your solution in AP Activity 4.2.2? Do you consider the solution better or worse? In what ways?