

Problem Statement: Deck of Cards

Code in an object-oriented programming language a collection of classes to model a deck of cards. These are poker-style playing cards (fifty-two cards in four suits: hearts, spade, clubs, and diamonds, with a face value of Ace, 2-10, Jack, Queen and King).

One of your classes should contain a `shuffle` method and a `deal_card` method.

| |
|--|
| <code>shuffle()</code> |
| Shuffle returns no value, but results in the cards in the deck being randomly permuted. Do not use a library-provided shuffle function. You may use library-provided random number generators. |

| |
|---|
| <code>deal_card()</code> |
| This function should return one card from the deck to the caller. |

A call to `shuffle` followed by 52 calls to `deal_card` will result in the caller being provided 52 cards of the deck in random order. If the caller makes a 53rd call to `deal_card`, no card is dealt.

There are many details to this task left vague on purpose. You should follow the principles of least surprise in making design decisions for this implementation.

While this is a trivial task, pretend that this code will be used as a basis for a new product. Take whatever measures you feel are necessary to meet this bar. Take your time with this assignment and “do it right.”