

```
//
// Deck.m
// Matchismo
//
// Created by abruzzim on 12/1/14.
// Copyright (c) 2014 Framework Solutions, LLC. All rights reserved.
//

#import "Deck.h"

// Private declaration

@interface Deck()

// Private storage within which to keep each card array pointer.
// Make space in the instance for the pointer "cards".

@property (strong, nonatomic) NSMutableArray *cards; // Collection of cards.

@end

@implementation Deck

// Allocate heap storage in the "getter" for the "cards" @property.

- (NSMutableArray *)cards
{
    // If the cards pointer is nil then alloc and init or
    // "Lazy Instantiate" the array.

    if (!_cards) {
        _cards = [[NSMutableArray alloc] init];
    }

    // Return an empty mutable array.

    return _cards;
}

- (void)addCard:(Card *)card atTop:(BOOL)atTop
{
    if (atTop) {
        [self.cards insertObject:card atIndex:0];
    } else {
        [self.cards addObject:card];
    }
}

- (void)addCard:(Card *)card
{
    [self addCard:card atTop:NO];
}

- (Card *)drawRandomCard
{
    Card *randomCard = nil;
}
```

```
// If there is at least 1 card in the array
// then return a random card. The "if" logic
// prevents an array index out of bounds error.

if ([self.cards count]) {
    unsigned index = arc4random() % [self.cards count];
    randomCard = self.cards[index];
    [self.cards removeObjectAtIndex:index];
}

return randomCard;
}

@end
```