

INTRODUCTION TO COMPUTER SCIENCE I

PROJECT 7 Poker Hands

Revision 0¹

1 Some new object types

For this assignment, you are given one new object type (the `Card` class), and a program (`PokerTest`) that expects another new object type (the `Deck` class) that **you must write**. Specifically, it is written to *create* a new `Deck` object—which itself is supposed to create one of each of the 52 possible `Card` objects—and then call on that `Deck` object to *shuffle* its `Cards`. Then, it *draws* `Cards` from the `Deck`, one at a time, to make poker hands. It tests each poker hands, counting up instances of different types (4 of a kind, full house, etc.).

The final result is that `PokerTest` will print, after generating 100,000 poker hands in this manner, the number of each type of hand that it observed (e.g., how many flushes). It will print this information alongside expected numbers of each type of hand. **If your shuffling method is a good one** that randomly permutes the cards well, then the results found will be near the expectations.

2 Your assignment

2.1 Copying files from my directory

You must copy some files from my directory to get started on this project. To do so, follow these steps:

1. Login to `remus/romulus` and open a shell.
2. Create a `project-7` directory and change into it.
3. Issue the following command from within your `project-7` directory:

```
cp -r ~sfkaplan/public/cs11/project-7/* .
```

If you look at your directory (use the `ls -l` command), you will see that there are two Java source code files:

- `Card.java`: A class that defines the contents of a `Card` object. Used to represent a single, standard playing card, each of which has a *suit* (*spade*, *club*, *diamond*, or *heart*) and a *rank* (2 through 9, *ace*, *jack*, *queen*, or *king*).
- `PokerTest.java`: The program, described above in Section 1, that repeatedly generates `Decks` of `Cards` to make poker hands and test their contents.

¹See Appendix A for a revision history.

2.2 The overall goal

You need to write the `Deck` class to make this program work. Specifically, a `Deck` object must contain an **array of pointers** to `Card` objects. Moreover, you must write the following methods:²

1. *Constructor*: Create one of each of the 52 possible playing cards—each possible combination of *suit* and *rank*—and keep pointers to each such card in the array of `Card` pointers. To perform this task, open the `Card.java` file, and notice the two arrays named `Card.suitNames` and `Card.rankNames`. Each is an array of pointers to `String` objects, each of which respectively contains a valid suit or rank name. Moreover, there is a `Card` constructor that accepts a suit name and a rank name as arguments. Loop over those arrays, calling the constructor with each possible combination of suit and rank, to make the 52 cards.
2. `shuffle`: Randomly permute the array of `Card` pointers.
3. `drawTopCard`: Assuming some order on the array of `Card` pointers, return the first such pointer, being sure to assign the array entry from which the pointer was taken to be `null`, thus “removing” that `Card` from the `Deck`.

3 How to submit your work

Use the `cs11-submit` command:

```
cs11-submit project-7 Deck.java
```

This assignment is due on Friday, May 7, at 11:59 pm

A Revision history

- **Revision 0 [2010-May-05]**: The initial, complete version.

²Determine the details of the *method signature*—the *static/non-static* designation, the *return type*, and the *parameter list* that accompany the method name—from the portions of the `PokerTest` code that calls `Deck` methods.