Lesson 89: Elevens Lab #2 (W31D2)

Balboa High School

Michael Ferraro

April 5, 2016

Do Now

- Take out the HW to be checked (questions 1–3 on p6)
- Make sure your `Deck` class functions properly with this code in `DeckTester`'s `main()`:

```
String[] ranks = { "Q", "5", "J", "9" };
String[] suits = { "diamonds", "spades" };
int[] values = { 10, 5, 10, 9 };

Deck d1 = new Deck(ranks, suits, values);

while( ! d1.isEmpty() ) {
    System.out.println( d1.deal() );
}

//see some order of QD, 5D, JD, 9D, QS, 5S, JS, 9S
```

Aim

Students will reinforce OO programming principles and practice various tasks in Java via the College Board's *Elevens Lab*.

Activity 2: `Deck` Class

Solutions to questions 1-3 (p6):

A2Q1	A `Deck` is a *collection* of `Cards`; A `Deck` `HAS-A` (multiple) `Cards`.
A2Q2	`ranks.length` $\times$ `suits.length` $=$ 6 cards
A2Q3	see next slide...

Activity 2: `Deck` Class

Solutions to A2Q3:

```
String[] ranks =
  { "2", "3", "4", "5", "6", "7", "8", "9",
    "10", "jack", "queen", "king", "ace" };

String[] suits =
  { "spades", "hearts", "diamonds", "clubs" };

int[] pointValues =
  { 2, 3, 4, 5, 6, 7, 8,
    9, 10, 10, 10, 10, 11 };
```

Activity 3: Shuffling a Deck

- Read *introduction* and *exploration* sections on pp7-9, making sure to understand. . .
    - the pseudocode for *perfect shuffle*
    - why *selection shuffle* is inefficient
    - how *efficient selection shuffle* is like Selection Sort
- In Eclipse, remove `Activity02` from the build path and make `Activity03` a source folder

Activity 3: Shuffling a Deck

Work on exercises 1 & 2 on p9

- `perfectShuffle()` will require a temporary array that's a copy of the parameter `values`.
- Recall that 8 `perfectShuffle()`'s of a 52-card deck should restore the deck's starting state — so try setting `SHUFFLE_COUNT` to 9[1] and `VALUE_COUNT` to 52 and see if that holds true.
- Increase `VALUE_COUNT` to have larger arrays to sort, trying both ODD & EVEN array lengths.

[1]Since values in deck are printed after a shuffle, you need to print 9 times to see the result of 8 shuffles between first line and last.

Activity 4: Add `shuffle()` to `Deck`

- Read *introduction* section on p11
- In Eclipse, remove `Activity03` from the build path and make `Activity04` a source folder
- Work on exercises 1 & 2 on p11
  - don't use `cards.remove()` — `set()` and `get()` methods will suffice
  - exercise #2 will be finished for HW

HW

- Do Activity 3, questions 1 & 2 (p9) — type up your solutions since they'll be run in a tester class.
- Finish Activity 4, exercise 2 — make sure you populate the deck with 52 cards and they shuffle when `DeckTester` runs.