

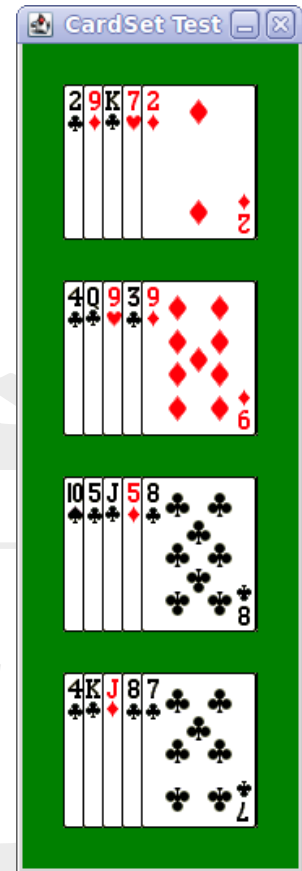
HW 2: CardSet Object

Create a `CardSet` object that contains many `Cards`. This represents a collection of `Card` objects: it could be as small as zero `Cards`, or as big as millions. This same object could be used to represent both a deck or a player's hand.

For this assignment, you'll have to use the `Card` object that you made during this week's lab. The `CardSet` object's primary field will be an `ArrayList<Card>`, which will actually hold the `Cards`.

You must create the following methods:

- `CardSet()`, the constructor which makes a new empty `CardSet`.
- `void add(Card card)`, which adds the given `Card` to the `CardSet`.
- `Card discard(int whichCard)`, that removes the `Card` at the given index and returns it.
- `void draw(Graphics2D pen, int x, int y)`, that draws the `Cards` in the `CardSet`. The `Cards` should overlap so that all but one `Card` have only their rank and suit visible.
- `int findHighCard()`, that returns the rank of the highest `Card` (aces low).
- `int findPairRank()`, that returns the rank of the highest pair of `Cards` with the same rank (aces low), or 0 if there are no pairs.
- `int findTripleRank()`, that returns the rank of the highest threesome of `Cards` with the same rank (aces low), or 0 if there are no triples.
- `int getSize()`, that returns the number of `Cards` in the hand.
- `boolean isFlush()`, that returns true iff all `Cards` in the `CardSet` are the same suit.
- `boolean isPair()`, that returns true iff there exist two `Cards` with the same rank.
- `boolean isStraight()`, that returns true iff all the `Cards` have consecutive ranks (such as 2-3, 9-10-J-Q-K, or even A-2-3-4-5-6-7-8-9-10-J-Q-K).
- `boolean isTriple()`, that returns true iff there exist three `Cards` with the same rank.
- `void shuffle()`, that shuffles the `Cards` into a random order.
- `void sort()`, that sorts all the `Cards` into the proper order. (It is easiest to make use of the `Collections.sort()` method.)



- `String toString()`, which returns a `String` that is a comma-separated list of the `Cards`, inside `()`s.

Please note that of these functions, only `add()`, `discard()`, `shuffle()`, and `sort()` should modify the ordering of the `Cards` in any way. All other methods should leave the `Cards` as they are.

In addition, there is a public static function that you must implement:

- `CardSet makeStandardDeck()`, which returns a brand-new `CardSet` that contains 52 `Cards`: one of each type. (This function will probably be the last thing you write. It will use some of the above functions.)

And at last, you should make a `main()` method for unit testing. This should create and draw four different hands of 5 `Cards`, which will be drawn one above the other. The `Cards` may be random or predefined as you like, but you should be sure to test all the methods you've made. (You should also have a standard deck, but you don't need to draw that.)

As with this week's lab, please try not to hard-code any values or file names. Use static final variables to avoid this.

The file you turn in should be called `CardSet.java`.

UNIVERSITY *of*
PUGGET
SOUND