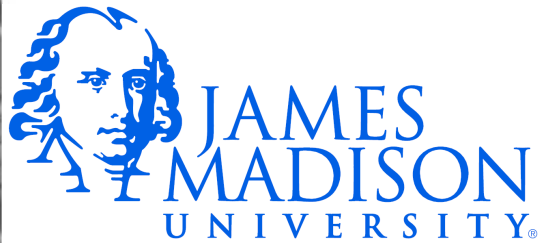




CS 149

Professor: Alvin Chao

CS149 – Arrays of Objects



More Classes and Objects

Color
<code>-red: int</code> <code>-green: int</code> <code>-blue: int</code>
<code>+Color(red:int,green:int,blue:int)</code> <code>+Color(other:Color)</code> <code>+add(other:Color): Color</code> <code>+darken(): Color</code> <code>+equals(obj:Object): boolean</code> <code>+lighten(): Color</code> <code>+sub(other:Color): Color</code> <code>+toString(): String</code>

Point
<code>-x: int</code> <code>-y: int</code>
<code>+Point()</code> <code>+Point(x:int,y:int)</code> <code>+Point(p:Point)</code> <code>+equals(obj:Object): boolean</code> <code>+getX(): int</code> <code>+getY(): int</code> <code>+setX(x:int): void</code> <code>+setY(y:int): void</code> <code>+toString(): String</code>

Classes generally include the following kinds of methods (in addition to others):

- constructor methods that initialize new objects
- accessor methods (getters) that return attributes
- mutator methods (setters) that modify attributes
- object methods such as equals and toString

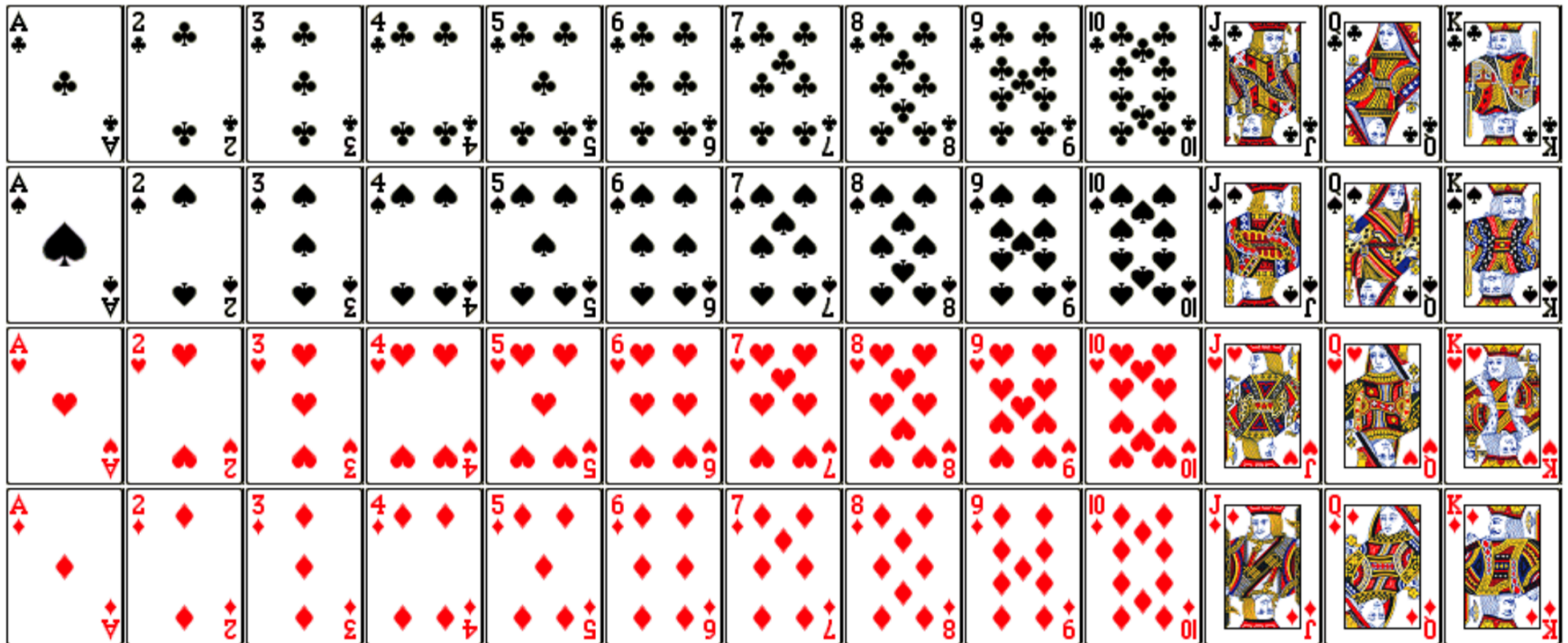


Identification

1. Identify the constructors for the Color class. What is the difference between them? What arguments do they take?
2. What kind of constructor does the Point class have that the Color class does not? What is the purpose of such a constructor?
3. Identify an accessor(getter) method in the Point class
 1. Which instance variable does it get?
 2. What arguments does the method take?
 3. What does the method return?
4. Identify a mutator(setter) method in the Point class.
 1. Which instance variable does it set?
 2. What arguments does the method take?
 3. What does the method return?
5. The Color class does not have accessors or mutators, but it provides methods that return lighter or darker Color objects. Why do you think the class was designed this way?

Model 2 Playing Cards

- Now let's play a different game. In this section, we'll create an array of face cards. (Note: The array is one-dimensional, but the cards are displayed on four lines for convenience.)





Card Questions

9. Implement the following constructor. The class has two attributes: rank and suit. Don't think too hard about it.

```
/**  
 * Constructs a face card given its rank and suit.  
 * @param rank face value (1 = ace, 11 = jack, 12 = queen, 13  
 = king)  
 * @param suit category ("clubs", "diamonds", "hearts", or  
 "spades")  
 */  
public Card(int rank, String suit) { }
```

10. In one line of code, declare an array of strings named suits and initialize its contents to the four possible suits as shown above.



Card Questions Contd.

11. Write several lines of code to declare and create an array of 52 Card objects. Use nested for loops to construct Card objects in the order of the Model. Make use of your suits array from the previous question. How you will keep track of the array index?

12. Describe what the following code does and how it works.(Note:You'vecomealongway this semester, to be able to understand this example!)

```
public static Card[] sort(Card[] deck) {
    if (deck == null) {
        System.err.println("Missing deck!");
        return null;
    }
    Card[] sorted = new Card[deck.length];
    for (Card card : deck) {
        int index = card.position();
        sorted[index] = card;
    }
    return sorted;
}
```

13. Write a static method named inDeck that takes a Card[] representing a deck of cards and a Card object representing a single card, and that returns true if the card is somewhere in the deck of cards. Be sure to use the equals method of the Card object to make comparisons.



- **Acknowledgements**

Parts of this activity are based on materials developed by Chris Mayfield and Nathan Sprague.

</end>