

Activity 12: Arrays of Objects

With arrays and objects, you can represent pretty much any type of data. It's not only possible to have arrays of objects, but also objects of arrays, objects of objects, arrays of arrays, arrays of objects of arrays, and so forth.

Model 1 Hand of Cards

Creating an array of objects is typically a 3-step process:

1. Declare the array

```
Card[] hand;
```

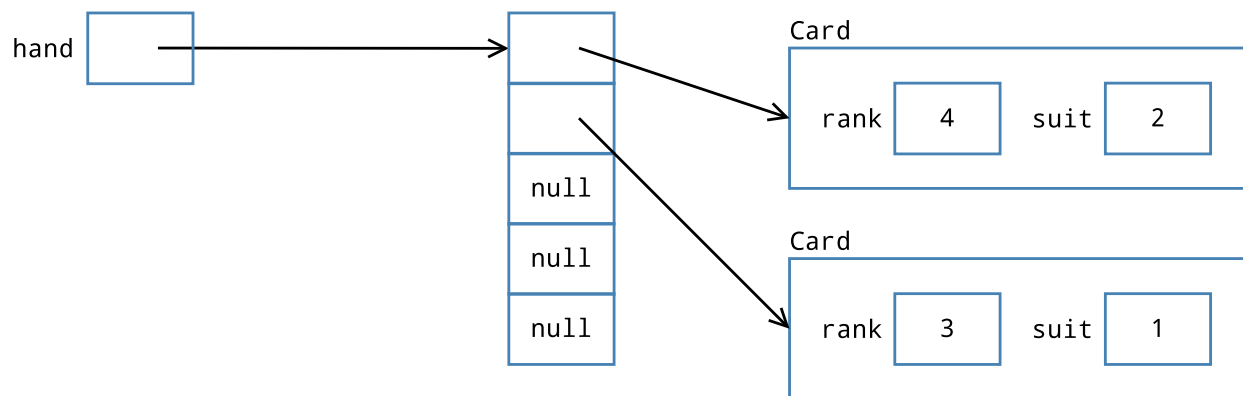
2. Instantiate the array

```
hand = new Card[5];
```

3. Instantiate each object

```
hand[0] = new Card(4, 2);
```

```
hand[1] = new Card(3, 1);
```



Questions (20 min)

Start time: _____

1. What is the type of the local variable `hand`? What is the value of `hand` *before* step 2? What is the value of `hand` *after* step 2?

2. When you create an array (e.g., `new Card[5]`) what is the initial value of each element?

3. When you construct a new object (e.g., `new Card(4, 2)`) what are the initial values of its attributes (e.g., `this.rank`)?

4. Describe in your own words what the following statement does. Be sure to explain how the random part works.

```
hand[(int) (Math.random() * hand.length)] = null;
```

5. What is the result of running the loop below? What is the purpose of the nested if-statement?

```
for (int i = 0; i < hand.length; i++) {  
    if (hand[i] != null) {  
        int suit = hand[i].getSuit();  
        System.out.println("The suit of #" + i + " is " + Card.SUITS[suit]);  
    }  
}
```

6. The *enhanced for loop* allows you to iterate the elements of an array. Another name for this structure is the “for each” loop. Rewrite the following example using a standard for loop.

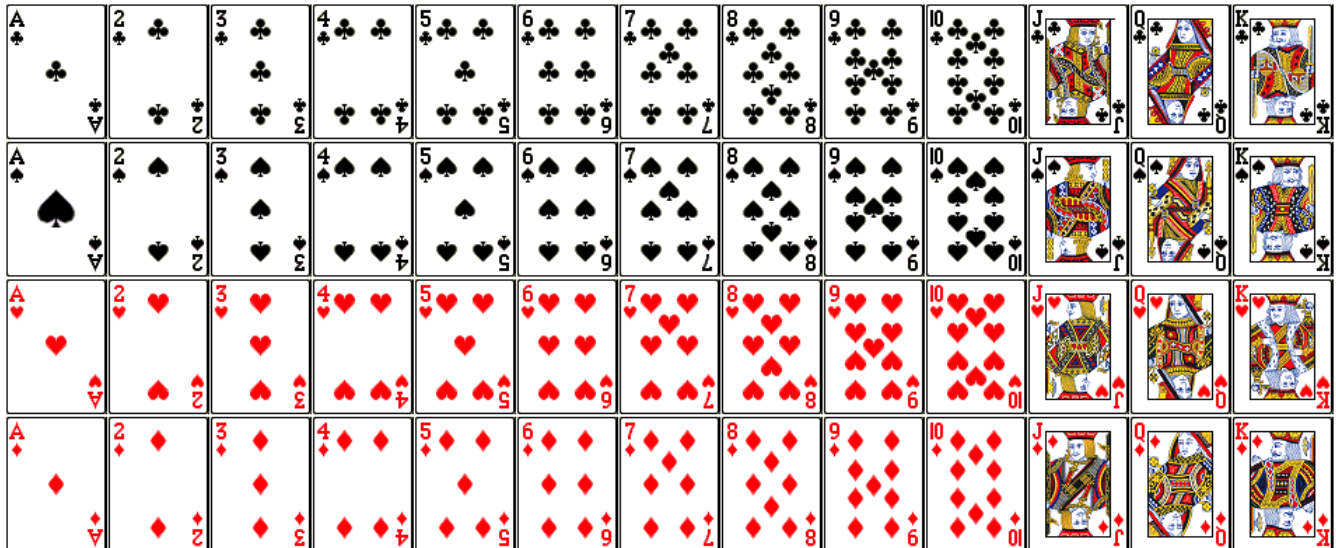
```
String[] days = {"Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat"};  
for (String day : days) {  
    System.out.println(day + " is a great day!");  
}
```

7. In contrast to enhanced for loops, what does a standard for loop typically iterate? Why would it be misleading to name the enhanced for loop variable `i` instead of `day`?

8. Rewrite the loop in #5 using an enhanced for loop. Use an appropriate variable name for the Card object (i.e., not `i`). For simplicity, you may omit the `System.out.println` line.

Model 2 Deck of Cards

There are 52 cards in a standard deck. Each card belongs to one of four suits (Clubs, Diamonds, Hearts, and Spades) and one of 13 ranks (Ace, 2, 3, 4, 5, 6, 7, 8, 9, 10, Jack, Queen, and King). The array below is one-dimensional, but the cards are displayed on four lines for convenience.



Questions (20 min)

Start time: _____

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.

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. Discuss with your team how you will keep track of the array index.

12. Describe what the following code does and how it works. (Note: You've come a long way 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();           // returns suit * 13 + rank - 1
        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.