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.

Manager: Recorder:

Presenter: Reflector:

Content Learning Objectives

After completing this activity, students should be able to:

- Explain the difference of instantiating an array and an object.
- Rewrite a for loop (over an array) using an enhanced for loop.
- Use enhanced for loops to construct and search arrays of objects.

Process Skill Goals

During the activity, students should make progress toward:

- Developing algorithms for constructing and searching arrays. (Problem Solving)

Model 1 Hand of Cards

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

1. Declare the array

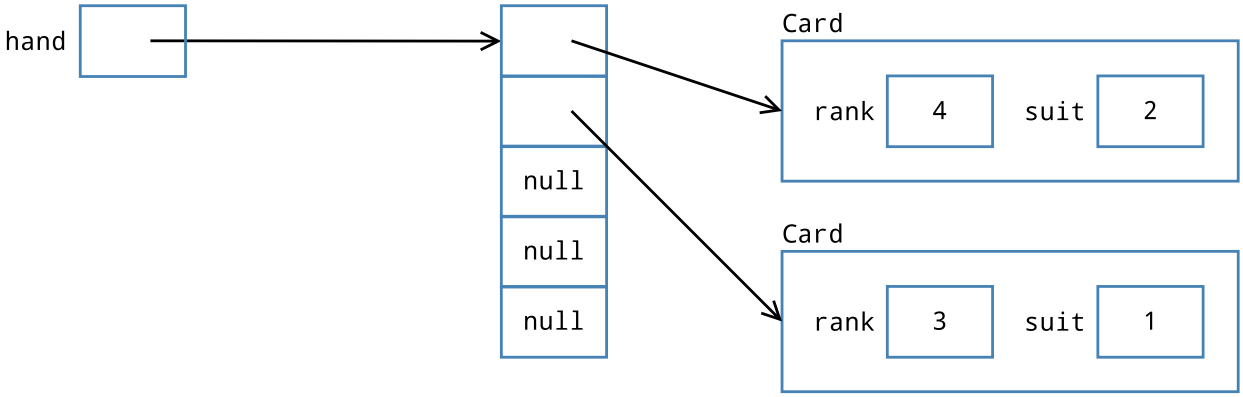
```
Card[] hand;
```

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`)?

The `new` operator finds a memory location to store an array or object. Java automatically determines how much memory is needed and initializes the contents of the corresponding memory cells to zero. That's why array elements and object attributes have default values, whereas local variables (not allocated with `new`) must be initialized before they are used.

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

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

5. What is the result of running the loop below? Explain why the if-statement is necessary.

```
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!");
}
```

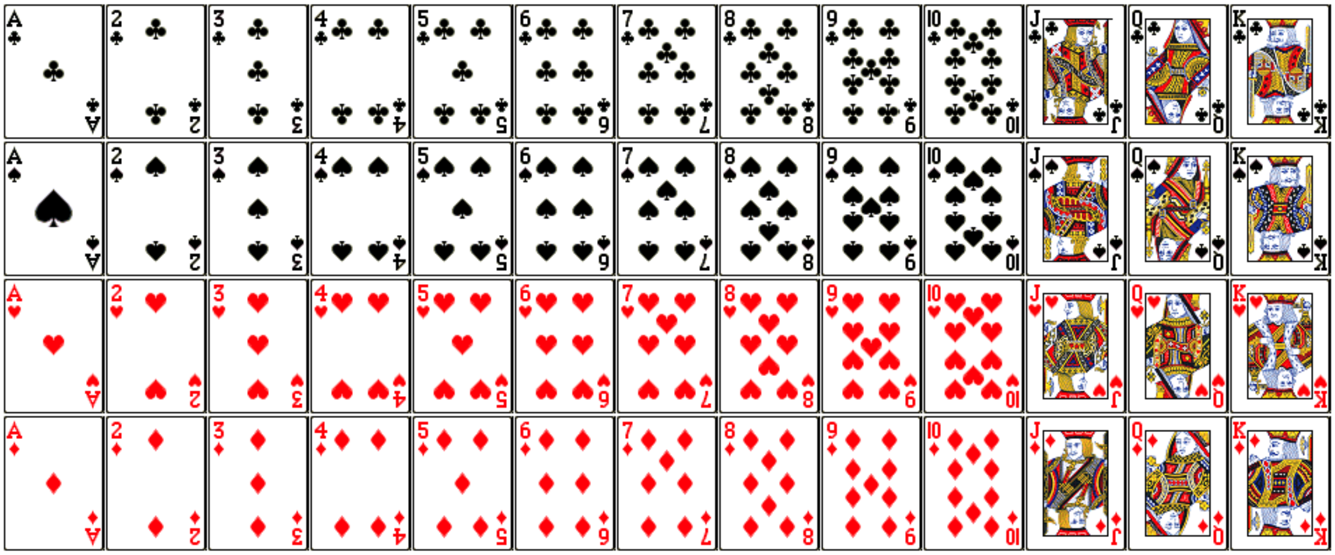
7. In contrast to enhanced for loops, what does a standard for loop iterate? Why would it be misleading to name the enhanced for loop variable `i` instead of `day`?

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 has one of ***13 ranks*** (1=Ace, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11=Jack, 12=Queen, and 13=King) and one of ***4 suits*** (0=Clubs, 3=Spades, 2=Hearts, and 1=Diamonds). For example, `new Card(12, 2)` would construct the Queen of Hearts.

The following deck is represented by an array of `Card` objects. The array is one-dimensional, but the cards are shown in four rows (because of the paper margins).



Questions (25 min) Start time:

9. What is the index (in the array above) of the following cards?

a) Ace of Clubs

b) Jack of Clubs

c) 2 of Spades

d) Queen of Spades

e) 7 of Hearts

f) King of Diamonds

10. Write the following statements using one line of code each.

a) Declare and initialize a `Card` array named `deck` that can hold 52 cards.

b) Construct the Ace of Clubs, and assign it as the first element in `deck`.

c) Construct the King of Diamonds, and assign it as the last element in `deck`.

11. Describe how you could repeat code from the previous question to construct the entire deck of cards (without having to type 52 statements).

12. Discuss the following code as a team:

```
int index = 0;
int[] suits = {0, 3, 2, 1};
for (int suit : suits) {
    for (int rank = 1; rank <= 13; rank++) {
        deck[index] = new Card(rank, suit);
        index++;
    }
}
```

a) What is the overall purpose of the code?

b) Why is the `suits` array not just {0, 1, 2, 3}? (See Model 2.)

c) Why does the code use an enhanced for loop for `suit`?

d) Why does the code use a standard for loop for `rank`?

e) What is the purpose of the `index` variable?

13. Write a 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.

14. 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;
}
```

a) What is the overall purpose of the code?

b) What is the purpose of the if statement?

c) Does this method modify the `deck` array? Justify your answer.

d) How does the `sort` method know where to put each card?

15. Identify the following Java language features in the previous question.

a) variables

b) decisions

c) loops

d) methods

e) arrays

f) objects