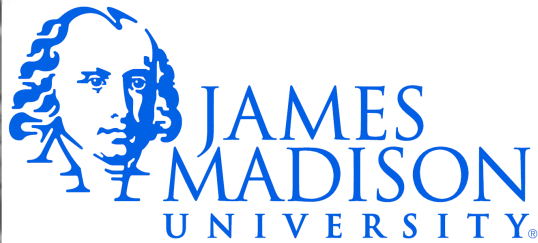




# CS 149

Professor: Alvin Chao

# CS149 – Arrays of Objects

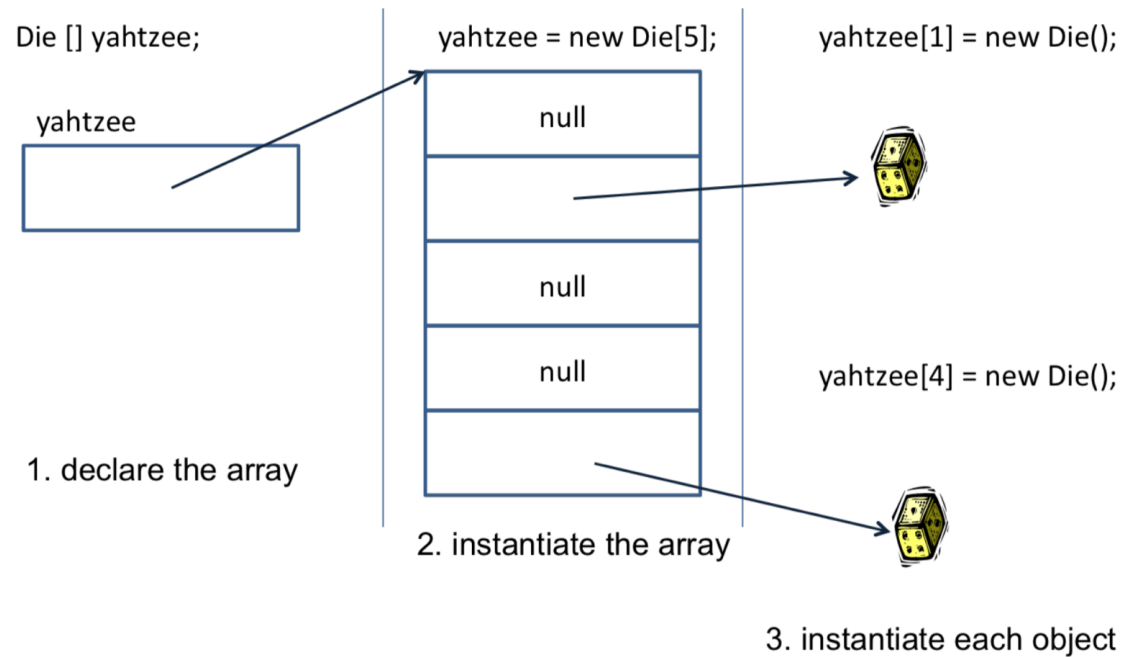


# Model 1 Yahtzee Dice

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 Yahtzee Dice

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





# Questions

- 1. What is the type of the local variable `yahtzee`? What is its initial value? (Hint: it's not null.)
- 2. When you create an array (e.g., `new Die[5]`) what is the initial value of each element?
- 3. When you construct a new object (e.g., `new Die()`) what are the initial values of its attributes (e.g., `this.face`)?



# Questions Cont'd.

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

```
yahtzee[(int) (Math.random() * yahtzee.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 < yahtzee.length; i++)  
{  
    if (yahtzee[i] != null) {  
        yahtzee[i].roll();  
    }  
}
```

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.

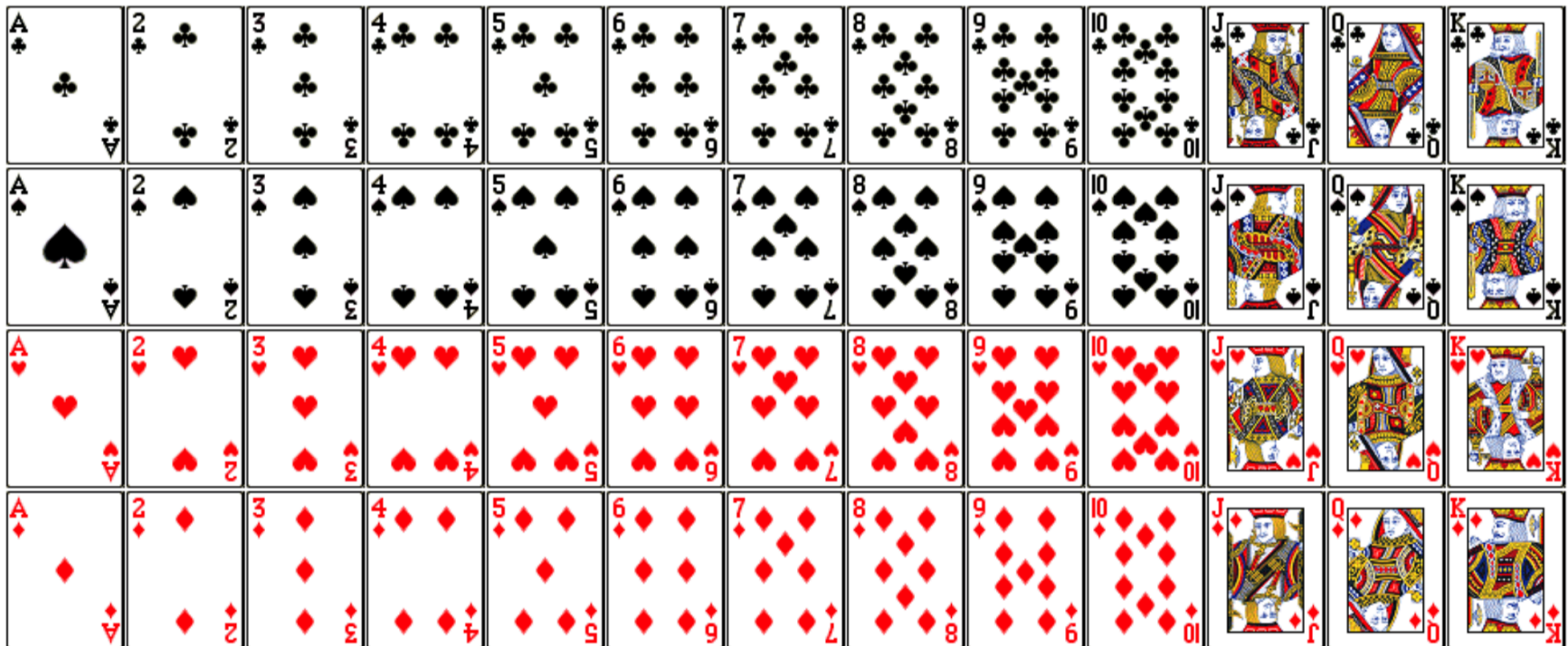
```
String[] days = {"Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat"};  
for (String day : days) {  
    System.out.println(day + " is a great day!");  
}
```

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 Die object (i.e., not *i*).

# 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've come along 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();
        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>