# Lab06<T>

Due: Fri May 8, 2015

Let's code up some generic stuff, ala Ch 17 Generics. This lab covers:

- ❏ Chapter 17 Generics
- ❏ Arrays.toString()
- ❏ Comparable interface

Copy my `Lab06.java` to get started. Hello, Lab06!

## 1. The Shuffler!

Write a class called `TheShuffler`. It should contain one **generic method** that randomly mixes up the order of the elements in an array of objects. Make the method `static`.

Add some code to main() to test your generic shuffle method on three array types: `String`, `int`, and `Integer`.

Similar to `Collections`, the `Arrays` class in the Java library is quite handy. Use the method `Arrays.toString()` to fancily print your results. For example, here are my results:

```
*** Test TheShuffler ***

Test String array
        Original=[ProfBill, AllyA, ConnorC, DwayneD, MattM, SteveS, WilliamW]
        Shuffled=[ConnorC, MattM, SteveS, DwayneD, WilliamW, AllyA, ProfBill]
        Shuf again=[MattM, SteveS, DwayneD, WilliamW, ConnorC, ProfBill, AllyA]

Test Integer array
        Original=[2, 3, 5, 7, 11, 13, 17, 19]
        Shuffled=[3, 7, 11, 17, 19, 2, 5, 13]
        Shuffled again=[7, 3, 19, 5, 17, 13, 11, 2]
```

Quiz - Is `Arrays.toString()` a generic method?
Look it up in the Java library Javadoc and let me know. Why did the Java library make this choice?

## 2. CompareDude

Create a **generic class** `CompareDude`. The type parameter for the dude is constrained to be any type that implements `Comparable`.

Remember that the `Comparable` interface defines one method: `compareTo()`. The `String` class is `Comparable`. I always remember it this way… x.compareTo( y) returns x - y.
Here's the Javadoc:

[docs.oracle.com/javase/7/docs/api/java/lang/Comparable.html](docs.oracle.com/javase/7/docs/api/java/lang/Comparable.html)

So, write two methods for the dude… (both will use `compareTo()`)

- `numOccurrences` - has two parameters (an array and an object) and returns the number of times the object occurs in the array
- `max` - has an array as a parameter and returns the largest object in the array.

Hrmph. What does a generic class and method signature look like. Consult the oracle, aka Ch 17, to figure this out.

For grins, here's my output. Your mileage may vary.

```
*** Test CompareDude! ***

Test String array:
        array1=[ProfBill, AllyA, ConnorC, DwayneD, MattM, SteveS, WilliamW]
        max=WilliamW
        numOccurrences(Bob)=0
        numOccurrences(AllyA)=1

Test Integer array:
        array3=[2, 3, 5, 7, 11, 13, 17, 19]
        max=19
        numOccurrences(161)=0
        numOccurrences(17)=1
```

## 3. Javadoc me

Once you're done and happy, clean everything up. Make it beautiful. Everything is **green**. Source/Format. Inline comments. Good **Javadoc** comments for all classes and methods!

Finally, use Netbeans to create web pages from your Javadoc comments, just like the Java standard library. It's the `Run/Javadoc` menu at the top. Your pages will be in the `dist/javadoc` folder of your NetBeans project. Look at them in your browser. (nice)

Done!
thanks… yow, bill