The contents of this document represent and go over the topics covered in lecture on 4/25 and 4/27 Use this to review the content as you please, also be sure not to use these in place of lecture as these documents will only cover lecture highlights and important details and do not represent the full scope of what you may be tested on. As always if you have any questions ask on ED!

Lecture 27 - 4/25/2023

Today we are going to continue our discussion on generics and also briefly discuss the Collections framework that is present in Java.

Recall from last time that we can create parameterized classes and we can even use generic parameters for our methods. Recall the binary search method you implemented for the last homework:

```
public static <T extends Comparable<T>> int binarySearch(T[] a, T x){
     //whatever code you decided to put here
}
```

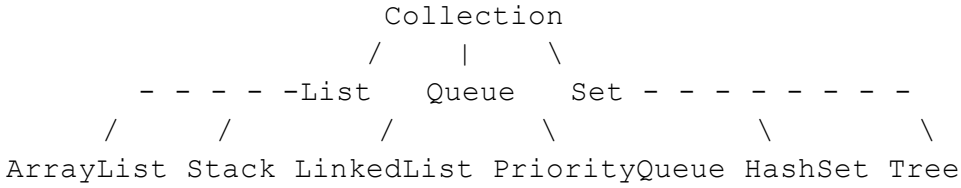
The clause between the static keyword and the return type acts as a restriction on the types T that we can pass into the method, in this case we are making the restriction that the type T has a properly implemented/overridden compareTo method meaning that types need to have some explicit relationship with T and Comparable<T>. At this point you should be very comfortable with everything else you see at this point, if not I highly recommend reviewing using the past materials. Java developers have an established convention with regards to generics that you should ideally be familiar with, I say ideally because *I* want you to know it, you won't be tested on this with 99% certainty (but hey Professor Cannon has been inspired in the past haha). This is the typical convention that I have been using in my office hours and that other Java developers tend to use as well. You won't get a compiler error or anything but it is nice to know.

The most commonly used type parameter names are:

- E - Element (used extensively by the Java Collections Framework)
- K - Key
- N - Number
- T - Type
- V - Value
- S,U,V etc. - 2nd, 3rd, 4th types

Speaking of the Java Collections Framework, let's talk about it. You'll implicitly become more familiar with it as you progress into 3134 as it

contains Data Structures that you will become more very familiar with after completing that course.
The main interface in the framework is (hopefully not surprisingly) Collection from there other interfaces implement/extend Collection and it eventually leads to the classes you see. Here is the schematic that is presented in the codio lecture:



(This tree turned out better than I had anticipated my text art has improved)

The graphic in the codio lecture has an extra interface that is separate from this tree but it is still relevant and related to what I have displayed here and I highly recommend you take a look at that graphic for the full picture.

Lecture 28 - 4/27/2023

No extra content today, except for things pertaining to the final.
Congratulations you have completed 1004! (the lectures at least)