

## LAB 03: ArrayList Hierarchy

**Directions:** Implement the classes `NonNullArrayList` and `OrderedArrayList` with the following specifications. Use at least two test cases for each class to test your scripts.

### NonNullArrayList

- `NonNullArrayList` is-a `ArrayList` (`NonNullArrayList` inherits from `ArrayList`).
- `ArrayList` uses any type of data. You must implement that behavior in your `NonNullArrayList` by declaring your class like this:

```
public class NonNullArrayList<T> extends ArrayList<T>{
    }
```

`<T>` indicates the generic type that will be used.
- Define two constructors: the default (no arguments) and the constructor with the `initialCapacity` as an argument. This is a subclass, you should remember to call the superclass constructor.
- Override these methods: `add` and `set`. Raise an exception `IllegalArgumentException` when a null value argument has been received. If the argument received is not null, you must call the `add` method from the superclass.

### OrderedArrayList

- `OrderedArrayList` is-a `NonNullArrayList` (`OrderedArrayList` inherits from `NonNullArrayList`).
- Your code should allow `T` to use the `compareTo()` method which compares an object with another, and returns a numerical result based on the comparison. If the result is negative, this object sorts less than the other; if 0, the two are equal, and if positive, this object sorts greater than the other (Example: `o1.compareTo(o2)`)

```
public class OrderedArrayList<T> extends Comparable<T>>
    extends NonNullArrayList<T>{

    }
```

- Define two constructors: the default (no arguments) and the constructor with the `initialCapacity` as an argument. This is a subclass, you should remember to call the superclass constructor
- Override these methods: `add` and `set`.
- The `add` methods must place the new element in the correct location. If a null element is added, your code must add it anywhere so it throws an exception.

- The set method must remove the element at the target position and then call the method add to insert a new element which will throw the correct exception if the element is null (the exception should be the one thrown when super.add(null) is called).

**Note:** compareTo() will throw an exception (NullPointerException) if comparing a null value. This is not the correct exception to handle for this question.