CS2030 Programming Methodology
Semester 2 2021/2022

16, 17 February 2022
Problem Set #4

1. In the Java Collections Framework, `List` is an interface that is implemented by `ArrayList`. For each of the statements below, indicate if it is a valid statement with no compilation error. Explain why.

   (a)
   ```
   void foo(List<?> list) { }

   foo(new ArrayList<String>());
   ```
   ? tackles invariance

   (b)
   ```
   void foo(List<Object> list) { }

   foo(new ArrayList<String>());
   ```
   invariance

   (c)
   ```
   void foo(List<? super Integer> list) { }

   foo(new List<Object>());
   ```
   List is interface

   (d)
   ```
   void foo(List<? extends Object> list) { }

   foo(new ArrayList<Object>());
   ```

   (e)
   ```
   void foo(List<? super Integer> list) { }

   foo(new ArrayList());
   ```
   Raw types

2. The following static generic method `max3` that takes in an array of generic type `T` that such that `T` implements the `Comparable` interface.

   ```
   static <T extends Comparable<T>> T max3(T[] arr) {
       T max = arr[0];
       if (arr[1].compareTo(max) > 0) {
           max = arr[1];
       }
       if (arr[2].compareTo(max) > 0) {
           max = arr[2];
       }
       return max;
   }
   ```

   What happens if we replace the method header with each of the following:

T max = arr[0]
T is not a Comparable<T>

(a) `static <T> Comparable<T> max3(Comparable<T>[] arr)`

T is not a Comparable<T>
Need to typecast

(b) `static <T> T max3(Comparable<T>[] arr)`

(c) `static Comparable max3(Comparable[] arr)`

T change to Comparable

3. Suppose a `Fruit` class implements the `Comparable` interface, and `Orange` is a sub-class of `Fruit`, how would you change the `max3` method header in question 2 such that the parameter type is `max3` is `List<T>` instead? You should aim to make the method as flexible as you can.

<? extends <T extends Comparable<T>>>
<T extends Comparable<? super T>>