**CS2030 Programming Methodology**
Semester 2 2022/2023

15 & 16 February 2023
Problem Set #4
**Java Generics**

1. Consider the following JShell program fragment.

```
jshell> ImList<Integer> list = new ImList<Integer>()
list ==> []

jshell> int one = 1
one ==> 1

jshell> Integer two = 2
two ==> 2

jshell> list = list.add(one).add(two).add(3)
list ==> [1, 2, 3]
```

   Which of the following code fragments will compile? If so, what is printed?

   (a) `for (Integer num : list) { System.out.print(num + " "); }`

   (b) `for (int num : list) { System.out.print(num + " "); }`

   (c) `for (Double num : list) { System.out.print(num + " "); }`

   (d) `for (double num : list) { System.out.print(num + " "); }`

2. For each of the code fragments below, indicate and explain the source of the error(s).

   (a)
```
List<? extends Object> list = new ArrayList<Object>()

list.add(new Object())
```

   (b)
```
List<? extends Object> list = List.of("abc");

list.add("def");
String s = list.get(0);
```

   (c)
```
List<? super Integer> list = new List<Object>();

list.add(new Object())
```

   (d) `List<? super Integer> list = new ArrayList<int>();`

   (e) `List<? super Integer> list = new ArrayList();`

   (f)
```
List<?> list = new ArrayList<String>();

list.add("abc");
```

3. In the lecture, we have seen the use of the `Comparator<T>` interface with the method specification `int compare(T t1, T t2)` that returns zero if `t1` and `t2` are equal, a negative integer if `t1` is less than `t2`, or a positive integer if `t2` is less than `t1`.

```
public interface Comparator<T> { // <T> declared with class scope
    int compare(T o1, T o2);
    ...
}
```

A generic method `T max3(T a, T b, T c, Comparator<T> comp)` can be defined in JShell as shown below. The method takes in three values of type `T` as well as a `Comparator<T>`, and returns the maximum among the values.

```
jshell> <T> T max3(T a, T b, T c, Comparator<T> comp) { // <T> declared with
   ...>     T max = a;                                   // method scope
   ...>     if (comp.compare(b, max) > 0) {
   ...>         max = b;
   ...>     }
   ...>     if (comp.compare(c, max) > 0) {
   ...>         max = c;
   ...>     }
   ...>     return max;
   ...> }
|  created method max3(T,T,T,Comparator<T>)
```

(a) Demonstrate how the `max3` method can be called so as to return the maximum of three integers $-1$, 2 and $-3$.

(b) Other than `Comparator<T>`, there is a similar `Comparable<T>` interface with the method specification `int compareTo(T o)`. This allows one `Comparable` object to compare itself against another `Comparable` object.

```
public interface Comparable<T> {
    int compareTo(T o);
}
```

As an example, since `Integer` class implements `Comparable<Integer>`,

```
jshell> Integer i = 1 // 1 autoboxed to an Integer and assigned to i
i ==> 1


jshell> i.compareTo(2) // 2 autoboxed to an Integer and passed to compareTo
$.. ==> -1
```

Let's redefine the `max3` method to make use of the `Comparable` interface instead.

```
<T> T max3(T a, T b, T c) {
    T max = a;
    if (b.compareTo(max) > 0) {
        max = b;
    }
    if (c.compareTo(max) > 0) {
        max = c;
    }
    return max;
}
```

Does the above method work? What is the compilation error?

(c) Does the following declaration of `max3` work?

```
<T> T max3 (Comparable<T> a, Comparable<T> b, Comparable<T> c)
```

(d) To restrict `T` to have the `compareTo` method, i.e. any class that binds to `T` must implement `Comparable`, we redefine the type parameter `<T>` to be `<T extends Comparable<T>>`.

```
<T extends Comparable<T>> T max3(T a, T b, T c) {
    T max = a;
    if (b.compareTo(max) > 0) {
        max = b;
    }
    if (c.compareTo(max) > 0) {
        max = c;
    }
    return max;
}
```

Demonstrate how the method `max3` can be used to find the maximum of three values $-1$, 2 and $-3$. Explain how it works now.