

CS2030 Programming Methodology
Semester 2 2023/2024

Week of 30 September – 4 October 2024
Problem Set #5
Java Generics

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

(a)

```
void foo(List<? extends Object> list) {  
    list.add(new Object());  
}
```

(b)

```
void foo(List<? extends Object> list) {  
    list.add("def");  
    Strng s = list.get(0);  
}
```

(c)

```
void foo(List<? super Integer> list) {  
    list.add(new Object());  
}
```

(d)

```
void foo(List<?> list) {  
    list.add("abc");  
}
```

2. A generic method is defined below. The method takes in three values of type `T` as well as a `Comparator<T>`, and returns the maximum among the values.

```
<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;  
}
```

- (a) Demonstrate how the `max3` method can be called so as to return the maximum of three integers `-1`, `2` and `-3`.
- (b) By replacing the values of type `T` with ones that also implement `Comparable<T>`, 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.

3. Suppose a `Fruit` class implements the `Comparable` interface, and `Orange` is a sub-class of `Fruit`.

```

class Fruit implements Comparable<Fruit> {
    public int compareTo(Fruit f) { ... }
}

```

```

class Orange extends Fruit { }

```

We define a generic `maxList` method that takes in a `List<T>` and returns the maximum element based on the definition of `Comparable<T>`. Does the following declaration of the method work?

```

<T extends Comparable<T>> T maxList(List<T> list)

```

Try it out by finding the maximum of a list of fruits or a list of oranges. How do you declare the method so that it works for both types of list? You should aim to make the method as flexible as you can.