CS-420-1: Object-Oriented Design
Fall 2019
Northeastern Illinois University
Homework #2: Due Thursday, 09/12/19 at 5:00 p.m.
Generics

- Download the files provided for you from the course website and unzip them into a folder.
- Similar to Homework #1, import the `homework3.project`.
- Other than to uncomment the tests, do not modify the code in `homework3.project` → `src` → `test` → `java`.
- Make sure that your test runner is set to Gradle.

Problem 1:

1. In `homework3.project` → `src` → `main` → `java`, create a java class named `ReverseListDemo`.
2. This class should have one `static` method named `reverseList`. The method should take an `ArrayList` of any type and return a new `ArrayList` (of the type passed in) with the elements in reverse order.
3. Uncomment the tests in the `ReverseListDemoTest` file and run them. If they both pass, then the method has been created correctly.

Problem 2:

1. In `homework3.project` → `src` → `main` → `java`, create a java class named `Tuple`.
2. The `Tuple` class should be a generic class with two private generic instance variables representing the first and second element of the tuple.
3. The `Tuple` class should have a constructor that takes two parameters and sets the generic instance variable.
4. Create getters for the instance variables. Is this class immutable?
5. There should be a `toString` method that prints out the String values of the generic instance variables in the following format: `( <firstStringVal>, <secondStringVal>`. Remember that generics are objects - how do you get the String representation of an object?
6. Uncomment the tests in the `TupleTest` file and run them. If they pass, then the `Tuple` class has (most likely!) been created correctly.

Extra Credit: More realistic

1. In the `Tuple` class, create a static method named `compare` that returns an `int` and takes two `Tuple` parameters.

2. The method should only take `Tuple` objects that are comparable (i.e. that are `Comparable`).

3. The method should return `1` if the first element of the first `Tuple` is "greater" than the first element of the second `Tuple` and if the second element of the first `Tuple` is "greater" than the second element of the second `Tuple`.

4. The method should return `-1` if the first element of the first `Tuple` is "less" than the first element of the second `Tuple` and if the second element of the first `Tuple` is "less" than the second element of the second `Tuple`.

5. Otherwise the method should return `0`.

6. You have been provided with a package named `geometric` that contains classes that implement the `Comparable` interface. These classes are used in the `ComparableTupleDemo` class (currently commented out) to showcase the usage of the `compare` method.

7. When you think you have created the `compare` method correctly, uncomment the code in the demo class and see if you get the output specified below.

8. Additionally, an extra two commented lines have been provided. When these lines are uncommented, they should be underlined in red in IntelliJ to indicated that they would not compile. If you have created the `compare` method and these lines do **not** show up as red, then the method has not been created correctly.

9. To receive extra credit, the `Tuple` class and the `compare` method must be created correctly.

```
(Rectangle 8.0, Circle 12.566370614359172)
(Circle 50.26548245743669, Rectangle 16.0)
(Rectangle 8.0, Rectangle 16.0)
(Circle 12.566370614359172, Circle 50.26548245743669)
-1
-1
0
0
1
-1
```

Submitting your homework to D2L

- In IntelliJ, under `File`, choose "`Export to Zip File`".
  Use the default name of `homework3.project.zip`.

- Submit this file to D2L.