

CPSC 250L Lab 9

Comparable, Equals, and Encapsulation I

Spring 2018

1 Introduction

The focus of this lab is to define *comparator* methods for a class. A comparator is a method that compares two objects of the same type and returns whether or not another object is “less than”, “equal to”, or “greater than” the object that called the method. Examples of comparators are Java’s `<=`, `==`, `>=`, `Comparable<T>.compareTo(T o)`, and `Object.equals(Object o)`.

2 Exercises

Fork and clone the `cpsc250l-lab09` repository from the CPSC 250 student group for this semester.

2.1 Person

In this exercise, you will create a class that implements the `Comparable` interface.

Exercise 1

© Christopher Newport University, 2016

Create a class called `Person` that implements the `Comparable` interface. The `Person` class should have a single field of type `String` that will store the `Person`'s name. Implement the following methods.

1. `public Person(String _name)`

This constructor should set the `Person`'s name to the `_name` parameter.

2. `public String getName()`

This method returns the `Person`'s name.

3. `public void setName(String _name)`

This method changes the `Person`'s name to the `_name` parameter.

4. `public Person copy()`

This method returns a new `Person` with the same name as this `Person`.

Override the following methods.

1. `public int compareTo(Person other)`

This method returns the lexicographical comparison of this `Person`'s name and `other`'s name.

2. `public boolean equals(Object other)`

This method returns whether or not `other` is the same as this `Person`. If `other.getClass` does not equal `Person`, return `false`. Otherwise, return whether or not their names are equal.

Test your code against `PersonTest.java`. Do **NOT** proceed to the next exercise until all tests pass.

Exercise 1 Complete

Run:

```
git add .  
git commit -m "Completed exercise 1"  
git push origin master
```

3 Common Mistakes

Some solutions to common mistakes are as follows.

1. To get the lexicographical comparison of two `String` objects, use the `String.compareTo(String)` method.
2. Two objects are equal if and only if `compareTo` returns 0.