Sam Lyons
Assignment 2
u0861439

This assignment is traditionally done with Pair Programming. Were there times you wish you had a partner to bounce ideas off of? We will ask you to compare your experiences with these first, "single" to assignments to the paired assignments later on in the semester. Please use your answer here for reference later.

Yes, I would have liked to have a partner for this assignment. It wasn't awful, considering there were no timing tests to be done and thus the workload was manageable for one person, but it certainly wasn't ideal.

Java's built-in classes Comparable and Comparator are both interfaces for doing comparisons among objects. What is the difference in the two interfaces? Give a situation when it is best to use each. Is it possible to change the extra features in LibraryGeneric such that the Comparable interface is used instead of Comparator? Why or why not?

When a class implements the Comparable interface, it must also implement a compareTo method which defines the "natural" ordering of that object. The method is contractually obligated to follow certain rules, such as having a 0 be returned for objects that are equal. A Comparator, however, is a class whose objects to be sorted do not need to implement the interface. It can be implemented by some third class. For instance, the OrderByAuthor, OrderByIsbn, and OrderByDueDate classes within LibraryGeneric implement this interface to do their sorting, while LibraryGeneric does not, even though it's the object that is being sorted. So, a Comparable object would be best when the object is comparing itself to some other object while a Comparator object is best when you're comparing two different objects.

No, I would say it's not possible to change the extra features in LibraryGeneric to accommodate a Comparable interface. This is due to the fact that a class that implements the Comparable interface only gets one compareTo method. Thus we would only be able to perform one kind of comparison with our LibraryGeneric class (from my understanding, at least). The Comparator interface gives us the power to do any number of comparisons, which is why it best suits our needs for this assignment, given that we have to perform three different types of comparisons.

Comment about the efficiency of your programming time. Did you utilize the time spent on this assignment effectively? How might it be improved?

I felt I was fairly efficient with the time I spent on this assignment. I felt my approach to testing could be improved though. I waited until after I had written all of my preliminary code to begin writing tests, which inevitably ended up with me having to revisit sections of code that I had written in order to refresh my memory on what exactly said code was doing. If I had just immediately written tests after completing each method, the time I spent looking over past sections of code could probably have been significantly reduced.

Reiterate why writing Generic code is important for this course. Phrase your answer in terms of Data Structures and Algorithms.

Writing Generic code is important for this course because very rarely will you see data structures and algorithms that are meant solely for a particular type. For instance, sorting algorithms should be able to handle any type you give it. That's just practical. Why write a sorting algorithm that can only sort Strings when you can write essentially the same algorithm using generics that's able to sort any type it's given? You're able to write significantly more versatile code with significantly more uses when using generics.

How many hours did you spend on this assignment?

I would estimate I spent around 10-12 hours on this assignment.