1. 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, it was a very big project to do on my own, and there were quite a few times when I ended up just staring at my code wondering what in the world is wrong with it. I think having someone to get a second opinion from would have come very much in handy, as well as speeding up the debugging process.

2. 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?

Comparable is generally used for natural ordering, and its compareTo() method is implemented inside the class that implements from it. Comparator's compare() method is not implemented inside a class, but exists outside of it. This allows for the classes in question to be sorted by multiple elements. It is best to use Comparable for single element, natural ordering sorts whereas Comparator should be used for multiple element sorts. It would be possible to use Comparable instead of Comparator with LibraryGeneric, but it is easier to use Comparator because this particular class required different sorting techniques to be implemented.

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

I started this project before having ever used Eclipse's debug feature. After learning about it in a lab, I realized that much of my time trying to pinpoint errors in my code could been saved if I had known about it earlier. Other than that I think I did a pretty good job utilizing my time.

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

It is important to use Generics because the desired program may require data structures that use a diverse range of data types. Without Generics the programmer would have to implement a different version of each data structure to specifically use one type of data, whereas utilizing Generics would cut this down to one data structure.

5. How many hours did you spend on this assignment?

About 8 hours.