Diego Baez Lopez
U0759247
Assignment 2

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.

- I have worked with partners on assignments previously and for this assignment I think there weren't too many situations where I needed to bounce ideas off a partner. I know that the assignments will get more difficult as we progress so for those instances I know having a partner will be very beneficial.

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?

- Learning about Java's two interfaces comparator and comparable provided a small amount of struggle initially but they proved to be rather straightforward after further investigation. Comparable is a built in to the class of that class. It compares the "natural" order of the objects it is comparing. Arrays.sort() is an example of such a comparable. This functionality is built into arrays and has a predetermined idea of what to look for when comparing. This is extremely useful for ordering and searching for items within an array or other corresponding object or class. Comparator is slightly different, it also compares objects but is defined in a separate class. The sorting logic is then determined by the user. This can happen based on the attributes of the objects and the goals of the user/company that might be asking for a specific parameter. Within our assignment ordering by ISBN, author, and due date were examples of this. Ideally when dealing with arrays of just numbers comparable is ideal because the attributes of the array are simple enough to be compared based on their "natural" order. This would be good for any sort of mathematical equation that would have to be done on that specific collection of data. A comparator was perfect for our implementation of the library code, because there were different attributes to each book within our library it allowed us to slice and dice the collection based on what was necessary. This would be especially useful when collect large amounts of data

and trying to determine different percentages, values, or order of the collection. Some companies might be interesting in demographic data and if they collection has the that specific attribute it would be easy to gain that vantage using a comparator. As far as I can tell I do not see any immediate opportunity to use a comparable over a comparator within our library script. I did run into some issues trying to implement a comparator and found that Type does not allow for a straightforward implementation of that. It would have involved checking instances and casting different types for each specific input or situation. Then declaring a new variable type to bring on the the comparator features. This could be useful at some point if some situation called for it but did not seem necessary for this assignment.

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 think I utilized the time allotted efficiently, I spent portions of the weekend working on a plan of attack. I also plan to use the weekend to start a greater portion of the assignment rather than leave it completely to the work week.

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

- As stated in the class lecture generics allow for type-checking at compile time instead of run-time. This is a huge advantage especially when using an IDE like Eclipse. The IDE lets the user know the code will not work and changes can be made before running the code and hitting a block. Also it Generic classes allow classes to specify type parameters and avoid significant amounts of type casting.

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

- I spent a total of fifteen hours on this assignment.