**Daniel Williamson**

**Project #1**

**10/4/16**

**CS 249**

To begin this project, I started with step #1. All I wanted to do was get the ComparableSortTests to run. I wasn’t concerned with passing any of the tests. I just wanted it to run. Once I had accomplished that, I immediately began to work on passing the tests. I referred to my book and found code for each of the 3 algorithms. Using each of the books sort methods for the three sorts I modified them to work in my favor. After a while, and a lot of surfing the web, I finally figured out what must be changed/modified. And I eventually passed all the tests from ComparableSortTests. As for big O notation, Bubble sort would be O(N2), because of the for loop. Selection sort, would be once again, O(N2) because of the two for loops. And lastly, if you couldn’t guess already, Insertion sort would take O(N2) because of the while loop.

Once I passed all those tests I went on to step #2. Which was rather simple. I created a new class called SortTimeClass to keep track of how long, in milliseconds, it took to sort the data that was provided randomly. I went with probably the easiest to code way, which most likely is not the most efficient/smartest route to go. My times are recorded below, once again, in milliseconds:

Next, I continued on to step #3. Which was getting the comparators to pass the tests. Once again, I began by getting the ComparatorTests to simple run. Once I was able to run the tests I began to add in the necessary code to pass said tests. I first started with the artist comparator. It was fairly easy, especially after finishing the sort tests. I used nested if statements to determine if order the object needed to be sorted in. The harder of the three tests had to be the HotAndNewComparator. It started with an if statement and 4 nested if statements. Also, it wanted some items in descending order, rather the ascending. Which was an easy fix. You could change the operation around, or switch the -1 and 1’s. And lastly the big O notation for the comparators are all O(1).

After the much easier comparator tests, I moved onto step #4. Which was even easier! I literally copy and pasted the sorting methods used previously in step #1 and changed a few different texts. And boom! Was all done. By far the easiest step of the whole project.

Lastly step #5, the first sort is using bubble sort and the HotAndNew comparator:



The second sort is using selection sort and Artist comparator:



The last sort was using insertion sort with the Chronological comparator:

