**Lab: 5**

**Name: Ryan Shaw**

**Registration number: 1605008**

|  |
| --- |
| **Problem statement(s)**  **EX1:** Find the maximum, minimum and average of a list, first using iterators and then again using for loops.  **EX2:** Create a new class and within that class a new method called ‘sharedItems’. This will take two lists and return the elements the lists have in common. An exception will be thrown if there are no common elements.  **EX3:** Create two new methods that sort a list. One should sort using comparable and the other using .sort. |

|  |
| --- |
| **Program description(s)**  **EX1:** I used iterators and for loops to find the min, max and average value. For min I stored the lowest value and compared it to each number while iterating. If the iterating number is smaller than the lowest value variable, then it will set iterating number to the lowest number variable. The same goes for the highest but I checked if the iterating value was larger. For the average value I summed up all the values and divided by the size of the list.  **EX2:** The ‘Duplicates’ class loops through each item in list 1, and while doing that, loops through each item in list 2. If the current values of list 1 and list 2 are equal, then it adds it to the common list. If this list is empty after it has finished iterating it will throw a ‘NoSharedItemsException’. In my opinion it is not suitable to throw an exception here because it is not an exceptional case. This could be done using an if statement instead.  **EX3:** Sorted ascending first using a while loop to go through the list size, incrementing ‘count’ each iteration. It then loops through then list size again at the same time incrementing ‘itemIndex’. If the element at the current ‘itemIndex’ is greater than the element at ‘itemIndex + 1’ then it swaps them around, it will swap every element where necessary and end up with a sorted list. The way I sorted ascending the second time was using collection .sort. |

|  |
| --- |
| **Test description**  Reordered the list to guarantee it is sorting correctly. Tried different common items in the ‘Duplicates’ class and no common items. |
| **Known bugs**  - |
| **Possible improvements**  Add sort descending. |
| **Comments**  Had some difficulty sorting with comparable but managed to eventually figure it out. |
| **Extra credit**  Added sorting in descending order in the ‘Duplicates’ class. |
| **References**  **-** |