**Lab: Lab 4**

**Registration number: 1703055**

|  |
| --- |
| **Problem statement(s)**  The purpose of this lab was to explore the Java Collections Frame work and develop code using its components. With this we made use of generics in order to have a method which could operate with various different types of objects. In particular we used generics in order to create a **deleteSmallest** method, to delete the smallest item from a list. Additionally, we learnt how to use an additional method to replace all negative values in a list with their positives. |

|  |
| --- |
| **Program description(s)**  In the first exercise smallest method is pre outlined for me and it is a method which is able to operate on various object types. All in order for comparison to be made to any object type in this case. With this I ran the supplied and observed how it all worked. The smallest method was completed using the *compareTo* method in order to find the smallest value base on returned +’ve or –‘ve value returned.  In exercise 2,  Experimented with the addition of items to test if the *smallest* method was indeed working.  Plus lines added showing test of **deleteSmallest** method – which was successful. Additionally I completed the *deleteSmallest* method which utilized the earlier *Smallest* method to find where the smallest item is in the list.  Exercise 3,  replaceNegList was implemented which uses a for loop to loop through every value in the supplied List and set the value as the positive value no matter what by utilising the Math.abs() method |

|  |
| --- |
| **Test description**  ***Exercise 1 & Exercise 2 together:***  Test 1: change the order of addElementStatements. Does the *smallest* method really work?  Result 1: The value returned is indeed the smallest value. So PASSED!  Test 2: Is the method successful with other data types?  Result 2: Passed was also successful with Float and Double types!  ***Exercise 3:***  Test 1: Change order of words being added, does it still return smallest?  Result: PASS – hello is still returned with correct position for where it is in the list now. |
| **Known bugs**  None |
| **Possible improvements**  None |
| **Comments**  To improve this lab I would make it clear how the **compareTo** method actually makes comparisons. In particular with strings, many may think it is comparing to find the shortest length string. It in fact compares strings based on the Unicode value of the characters in the string. |
| **Extra credit**  For exercise 2, I created an additional vector. Added in order to test my method with other data types. |
| **References**  None |