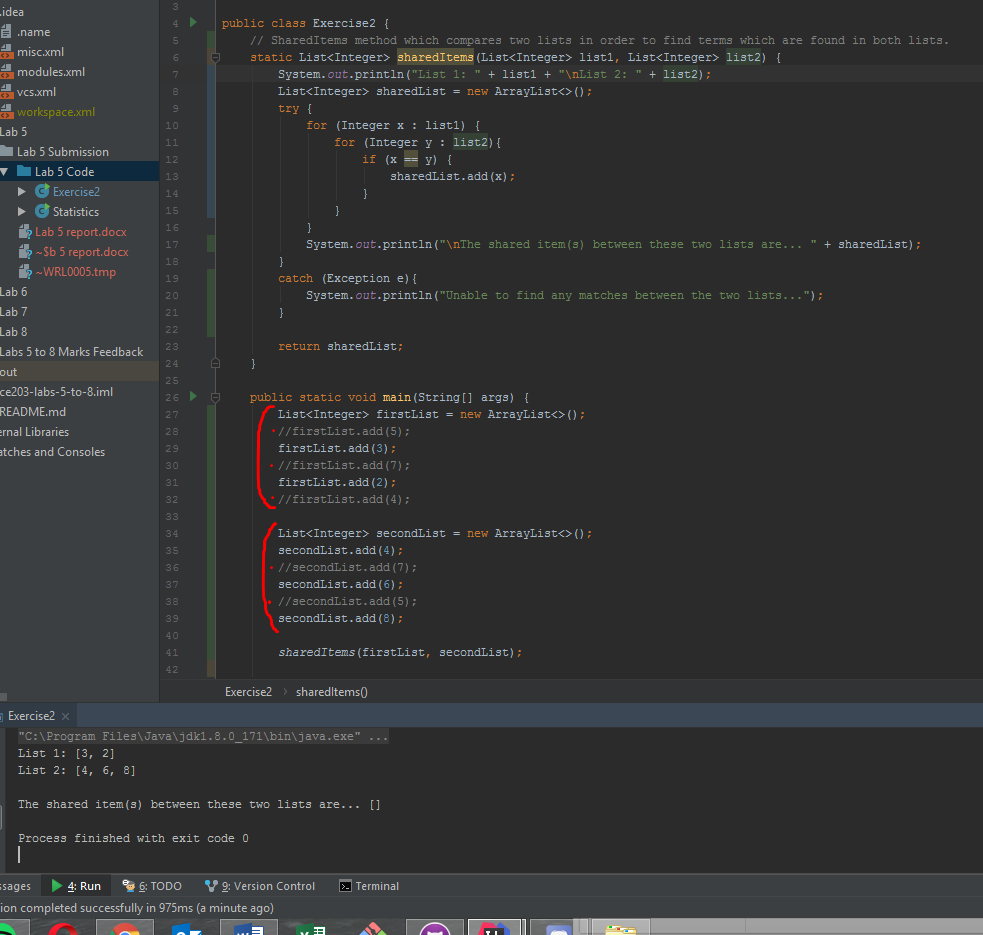
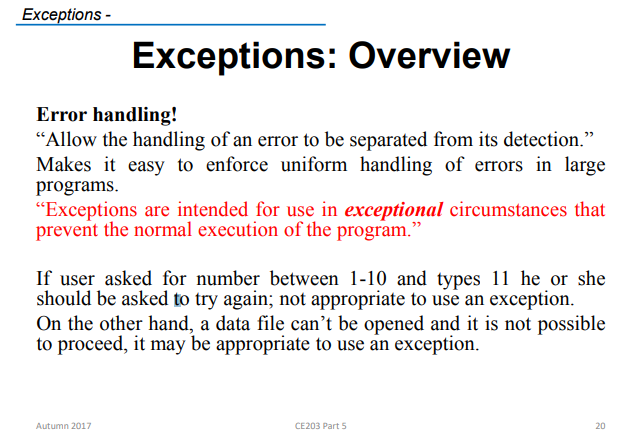
**Lab: Lab 5**

**Registration number: 1703055**

|  |
| --- |
| **Problem statement(s)**  Continuing with the Java Collection Framework and Exception handling. This helped build on our understanding of the data structures in order to use them properly. Further building on out class methods inorde to have specific methods to the relevant objects. |

|  |
| --- |
| **Program description(s)**  In the first exercise I implemented the three methods at first using iterators and then with for loops. Both were successful and it allowed my to understand how to implement it all in wo varying ways.  For the second exercise I successfully implementesd a shared method which looped through and found all shared items.  Exception handling in this case is not too appropriate as it would not be triggered with the method I used to find shared items. As I have used a nested for loop with an if conditional in order to find any matching Integers in the list, this can be seen in my code which adds matching items to the shared list. I tested if an error would be raised with a test set of two lists which would have no matching items. However as the below output shows, the exception handling part of my code has not been triggered. This may be because of the method I used to cycle through and see if there are matching items.  In order to catch lists where there are no matching Integers I would have used a better method of a simple *.size()* conditional in order to check if the sharedList is empty. If so an appropriate message would be shown.  **Please see to note in the possible improvements part of this report – I definitely should have used the Java collections *.retainAll()* method** |



|  |
| --- |
| **Test description**  Tested with various input strings of positive and negative numbers in different orders and it all ordered and made positive correctly. |
| **Known bugs**  None known |
| **Possible improvements**  For the second exercise I could have far improved it by using the java collections method of *.retainAll().* If I was redoing this exercise I would defiantly use this method instead of reinventing the wheel with my nested for loop approach. |
| **Comments**  none |
| **Extra credit**  Please see to possible improvements where I have discussed about the potential of making use of the java collections method of *.retainAll().* |
| **References**  none |