VennDiagram: Testing Document

For our Venn Diagram application, we tested all the methods in our 3 most important classes. These three important classes are our group, item and quantifying item classes. An exhaustive list and description of all test methods in these classes are as follows:

Group Tests:

- testGroupName()
  tests to see if getTitle method works correctly
  test was derived by
- testGroupName2()
  uses the setTitle method to change the title of the group, then runs the getTitle method to check if title has been changed correctly
- testGroupEmpty()
  tests to see if isEmpty method works correctly
- testGroupEmpty2()
  adds an item to a group and tests to see if isEmpty returns false
- testAddAndRemove()
  adds and removes an item from a group and then checks to see if the group is empty
- testAddAndRemove2()
  adds multiple items to a group and uses the removeAll method to get rid of them all; checks see if it is now empty
- testInsertItems()
  adds multiple items to a group and tests if they correctly inserted
- testRemoveItem()
  adds an item to a group and removes it
- testRemoveItems()
  adds multiple items to a group and tests removeItems method
- testRemoveAll()
  adds multiple items to a group and tests removeAll method
- TestGetItems()
  Tests the getItems method
- testContains()
  adds multiple items and checks to see if certain items are in the group
- testToVisualList()
  adds multiple items to a group and tests the toVisualList method
- testToSet()
  adds multiple items to a group, creates a set, tests to see if set is correct
- testFindMatching()
- tests to see if two groups contain any of the same items

Item Tests:

- testGetText()
- tests to see if an getTest method is working correctly
- testGetID()
  tests to see if the overridden hashCode method preforms correctly
- testIncrementID()
  tests to see if hashCode runs correctly for newly added items
- testSetText()
  changes the text of an item and tests if correctly changed
- testEquals()
  tests all possible inputs of the overridden equals method
- testToString()
  tests to see toString method is outputting correctly
- testHashCode()
  tests to see if the hashCode method is not zero.

QuantityItem Tests:

- testGetCount()
  tests to see if QuantityItem is initialized to a count of zero
- testIncrementCount()
  tests to see if incrementCount method works correctly
- testHashCode()
  tests to see if hashCode method is functioning correctly
- testEquals()
  runs most possible inputs of the overridden equals method

Our test coverage is 100% for both our Group and Item classes, and 90% for our QuantityItem class. This is to ensure that all methods within these classes are working effectively. By testing these main classes, we can ensure that our program can run effectively and precisely without any unwanted errors client side. By having a test coverage 100% can help ensure a fully functional program, free of unwanted bugs or errors. These specific classes make up the backbone of our application which is why we decided to cover them as much as possible.

These test cases were derived by ensure that our Venn diagram program would work correctly, even when given multiple items, or having no items whatsoever. This methodology of testing extreme individual scenarios ensures that our application can work for the needs of many customers, without them having to worry about bugs due to their input. These tests were also derived to ensure that we got ample test code coverage for our application. By covering as much of our code as possible, we can ensure that the application can run near-bug free for most users.

These test cases are sufficient because they have 100% test coverage for our group and item classes, and over 90% for our QuantityItem class. These tests also account for adding and removing multiple items at once, ensuring that our program can run smoothly. This means that our program can handle the needs of our customers without fail. These test cases therefore help ensure that our program can operate correctly, regardless of the information the user inputs. By testing methods in our item and group classes we can ensure the foundational structure of our Venn Diagram application runs perfectly.