CIS 120 Data Structures and Algorithms **University of Wisconsin-Stevens Point**

Lab 06: Extending the Generic List ADT **Due Date: as specified in the course schedule**

In this lab you will get the opportunity to create your own project using the ADT as presented in Chapter 6. Then you will add functionality to the ADT at the implementation level.

Brief Overview

1. Consider the operation *isEmpty* which returns true if the list is empty, false otherwise. Add this method to the ArrayUnsortedList class. Code and test (using JUnit testing) your method. The unit testing should include all possibilities for the list.

2. Consider the operation *removeAll* which removes all elements from the list that are equal to the argument element and returns an int indicating how many elements were removed. Design a method to be added to the ArrayUnsortedList class that implements the operation. Code and test (using JUnit testing) your method. The unit testing should include all possibilities for the list.

How to proceed

Create a new Java project, Lab06_ListADT. Create a package to contain the list classes (call it `lists`). Copy in the necessary list classes, starting with ArrayUnsortedList. (These files are all in the Chapter 6 source folder.) Create a Word doc containing all the test cases for the `isEmpty()` method. (Look back all the way to StringLog if necessary.) Then, write the `isEmpty()` method, followed by sequentially applying the test cases. (This is test driven development.) To test the method, add a testPackage package to the project. Add a `test_isEmpty` JUnit test case (class) to the package, and stepwise include all the possible list cases to confirm the correctness of your `isEmpty()` method.

NOTE: you must *import* the ArrayUnsortedList class with the following statement:

import lists.ArrayUnsortedList;

Do the same for the `removeAll()` method. Note that a properly tested `removeAll()` method will likely have from ten to twelve test methods.

What to hand in

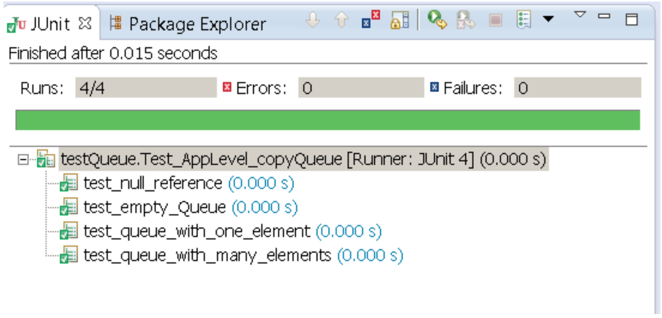
Upload the following files in order:

1. The Word doc containing the test cases, for both methods.
2. The java files ArrayUnsortedList.java, test_isEmpty.java, and test_removeAll.java.
1. The PDF files ArrayUnsortedList.pdf, test_isEmpty.pdf, and test_removeAll.pdf. PDF files are created by using Eclipse's ability to "Print to PDF".
   The PDF file can be created from within Eclipse by *printing* to PDF. File | Print | Microsoft Print to PDF |(supply file name)
   The PDF submission allows your instructor to add Canvas "Annotations" adjacent to the indicated Java program lines.

3. Screen shots of the testing window, copied into a Word doc, demonstrating successful completion of the tests.
   NOTE: use "Snip" (Windows), or pressing Shift-Command (⌘)-5 (Macintosh).

4. A Word doc containing a paragraph describing the big-O algorithmic analysis for each of `isEmpty()` and `removeAll()`. Pay particular attention to the impact upon the result by `removeAll()` perhaps repeatedly calling `remove()`.

   Here is an example of the snip of the testing window.



```
Finished after 0.015 seconds

Runs: 4/4     Errors: 0     Failures: 0

testQueue.Test_AppLevel_copyQueue [Runner: JUnit 4] (0.000 s)
    test_null_reference (0.000 s)
    test_empty_Queue (0.000 s)
    test_queue_with_one_element (0.000 s)
    test_queue_with_many_elements (0.000 s)
```