CS 1699 - FINAL DELIVERABLE

Writing a new app from scratch using TDD

Charlotte Chen and Daniel Hui

**What we tested?**

We decided to *write a new app from scratch using TDD.* TDD seemed like an interesting and effective approach to writing software. Writing the tests before the actual codemeant that we would hopefully spend less time debugging. Also, TDD would provide us with an entire test suite for our program to be used if the code is updated. This sounds like a very efficient, organized, and structured way to write code. We build a simulator program for a conga line implemented using a custom LinkedList class that we also built.

([*https://en.wikipedia.org/wiki/Conga\_line)*](https://en.wikipedia.org/wiki/Conga_line))*.*

**How?**

We started by building a Node class. Using this Node class we were able to build a LinkedList class that implemented the following functionalites: getLast(), size(), indexOf(), remove(), get(), add(Node), add(T value), addLast(), clear(), indexOf(), and addAtIndex(). Next, we built a real-life application implementing our LinkedList class to build a Conga Line simulator program

([*https://en.wikipedia.org/wiki/Conga\_line)*](https://en.wikipedia.org/wiki/Conga_line))*.*

This program had the following functionalities: person enters line, person enters behind a specific person, person leaves the line, everyone leaves the line, and show whose hips a person is holding.

**Testing Aspect?**

Before we any wrote any new function for a class, we wrote test code using JUnit that would test the function for errors or inconsistencies. There are two parts to our tests. The first JUnit test tests the methods in the LinkedList class and the second JUnit test tests the methods in the CongaLine class. Using this system, we would develop software that is properly tested, and also help us further understand the software we are writing.

**Problems/Difficulties**

Our LinkedList class throws an IndexOutOfBoundsException Exception if an the index passed to get() is not between 0 and the size of the LinkedList. So to test whether this situation is handled properly JUnit would need to check if an exception is thrown. Luckily, junit provides a functionality to test for an exception:

***@Test*** *(e*xpected= Exception.class)

It was sometimes difficult in envisioning every possible scenario/errors that we would have to test for while writing the tests without writing any actual code for the methods. We seemed to neglect some error handling cases, edge cases, and corner cases within the methods that we would need to test for. For instance, we wrote the test methods for the remove method in our LinkedList class. However, while writing the code for the remove method we realized that removing the first element in the list would be handled differently than

removing an element in the middle of the list. So using TDD, we sometimes needed to add more test code after realizing our code was not fully covered.

**Assessment of TDD**

Overall, we both enjoyed using the TDD way of writing software. We really felt that after writing test code we developed a more comprehensive and clearer sense of how we would write each method. We were a lot more confident while writing and more aware of what we had to do. Thus, this saved us time while actually writing the code. Just as importantly, if we wanted to update our software, we would already have a test suite for the entire system. Hopefully, we have the opportunity to write code using this system in our careers.

**Ready for Release**

The Conga Line Simulator implemented via a Linked List was written using Test-Driven Development. This means that all code in our Linked List and Conga Line programs have been tested for any possible defects. The Conga Line program and Linked List program both meet the basic functionalities without error. Currently, after fixing all defects we found that the both programs run smoothly. After rigorous testing and refactoring code, our software testing team has decided that this software is ready for release immediately.

[Mention code coverage]

**https://github.com/dth26/CS1632\_FINAL\_PROJECT**