Assignment 3

Objectives

- Further experience working with Objects and Arrays in Java
- Experience writing your own tests
- Experience implementing a doubly-linked list in Java

Objectives

This assignment requires you to implement a list interface using a doubly-linked list implementation.

Although there is a tester provided for this assignment, it does not include a comprehensive set of sets for each method. You should add your own tests to test cases not considered.

Note: The automated grading of your assignment will include different and additional tests to those found in the A2Tester.java file.

Quick Start

1. Download all of the .java files found in the Resources > Assignments > a3 folder.
2. Read through the documentation provided in ther `A3List.java` interface, there is a lot of very useful information that will help you with your implementation of each method.
3. Compile and run `A3Tester.java`. Work through implementing each method at a time. Debug the method until all of the tests pass for that method before proceeding to the next one.
4. The following video outlines what is expected of you for the interleave method: https://connex.csc.uvic.ca/access/content/group/6eb1a305-4430-415c-95dd-5907babf80fd/videos/html/interleave.html

CRITICAL: You **must** name the methods in A2Aligner.java as specified in the documentation and used in `A2Tester.java` or you will receive a **zero grade.** Remember that all methods specified in an interface must be implemented, or your code will not compile correctly. Any compile or runtime errors will result in a **zero grade** (as if the tester crashes it will not be able to award you any points for any previous tests that may have passed).

Submission and Grading

Submit **A3LinkedList.java** with your name and student ID at the top using conneX**.**

If you chose not to complete some of the methods required, you **must provide a stub for the incomplete method(s)** in order for our tester to compile. If you submit files that do not compile with our tester, you will receive a **zero grade** for the assignment. It is your responsibility to ensure you follow the specification and submit the correct files. Additionally, your code must **not** be written to specifically pass the test cases in the tester, instead, it must work on all valid inputs. We may change the input values when we run the tests and we will inspect your code for hard-coded solutions.

Be sure you submit your assignment, not just save a draft. **ALL late** and **incorrect** submissions will be given a **ZERO** grade. A reminder that it is OK to talk about your assignment with your classmates, but not to share code electronically or visually (on a display screen or paper). We will be using plagiarism detection software.