Assignment 5

Reminder: Your code is to be designed and written by only you and not to be shared with anyone else. See the Course Outline for details explaining the policies on Academic Integrity. Submissions that violate the Academic Integrity policy will be forwarded directly to the Computer Science Academic Integrity Committee.
All materials provided to you for this work are copyrighted, these and all solutions you create for this work cannot be shared in any form (digital, printed or otherwise). Any violations of this will be investigated and reported to Academic Integrity.

Objectives

- Practice implementing linked list data structure
- Practice implementing an interface
- Practice reading and understanding specifications

Introduction

This assignment will build on your knowledge of the list interface and introduce you to a reference-based implementation of a list as opposed to the array-based list you implemented in Assignment 4

Your assignment is to implement the `IntegerList` interface defined in `IntegerList.java` as a doubly-linked list in `IntegerLinkedList.java`

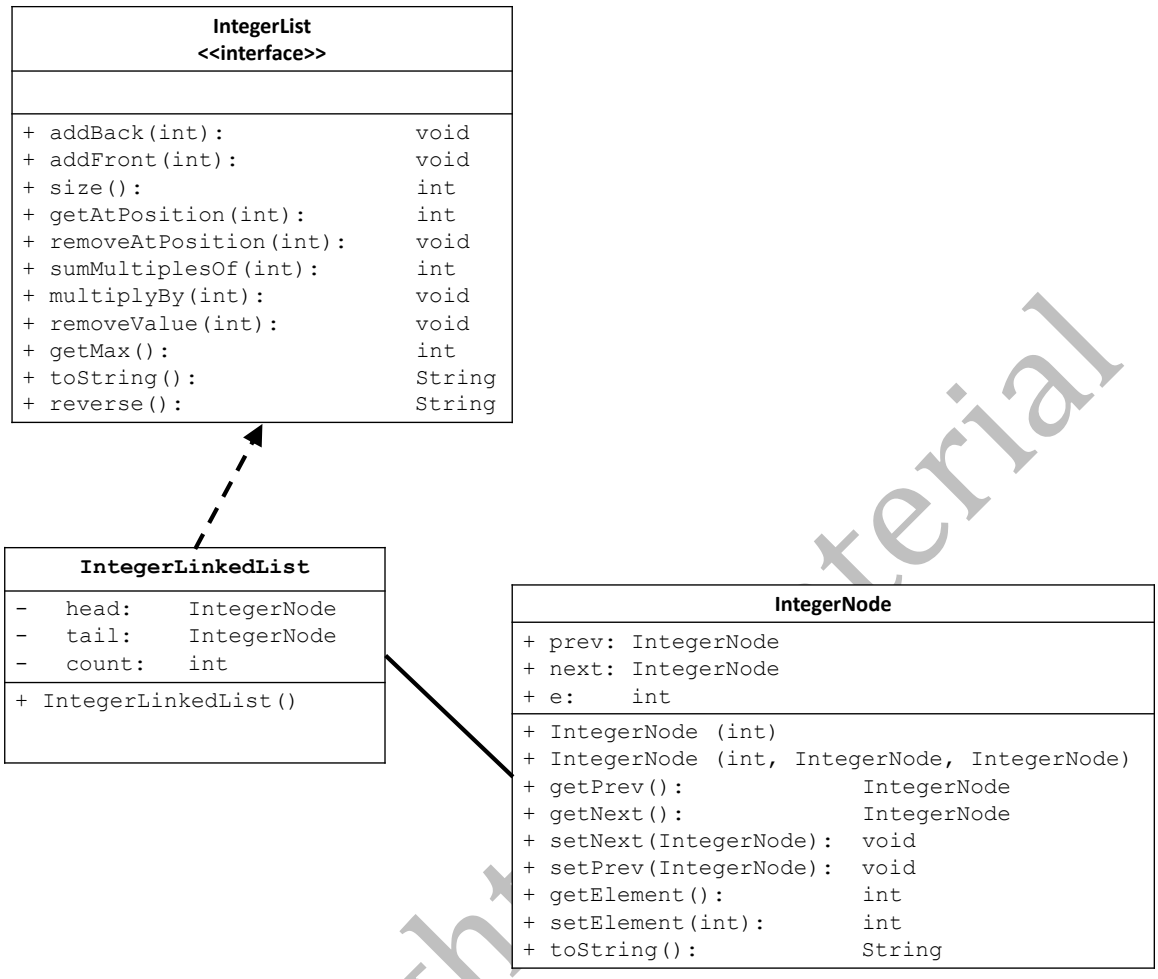
A UML overview is provided on the following page.

We have provided you with an implementation of the `IntegerNode` class. Ensure you read and understand this implementation so that you can use it correctly.

Submission and Grading

Submit your `IntegerLinkedList.java` with the completed methods through the assignment link in BrightSpace.

- You **must** name the methods in `IntegerLinkedList.java` as specified in the given interface and as used in `A5Tester.java` or you will receive a **zero grade** as the tester will not compile.
- If you chose not to complete some of the methods required, you **must at least provide a stub for the incomplete method** in order for our tester to compile.
- If you submit files that do not compile with our tester (ie. an incorrect filename, missing method, etc) you will receive a **zero grade** for the assignment.
- Your code must **not** be written to specifically pass the test cases in the testers, instead, it must work on other inputs. We may change the input values when we run the tests and we will inspect your code for hard-coded solutions.
- **ALL late** and **incorrect** submissions will be given a **ZERO** grade.



Getting Started

1) Download all java files provided in the Assignment link on BrightSpace.

2) Try to compile `A5Tester.java` You will see it does not compile because your `IntegerLinkedList` class is missing the required methods.
NOTE: we have provided you with the `toString` and `reverse` method implementations – DO NOT change these.

3) Introduce stubs for your constructor and for each of the methods `IntegerLinkedList` must implement.
DO NOT move on until you have the tester compiling with no errors!

4) Implement each method in `IntegerLinkedList.java` by repeating the following until all of the test methods in `main` of `A5Tester.java` are uncommented and all tests pass.

   a) Uncomment one of the test methods in the `main` of `A5Tester.java`

   b) Implement ***one*** of the methods being tested in `IntegerLinkedList.java`

   c) Compile and run the test program `A5Tester.java`