

Introduction

In this assignment, you will be using the `List` ADT to solve a number of problems in Part 1, and then will implement a generic stack in Part 2.

The automated grading of your assignment will include some different and additional tests to those found in the `A4Tester.java` file, as it does not include a comprehensive set of tests for each method. You are expected to write additional tests until you are convinced each method has full test coverage. The [displayResults](#) and [test coverage](#) videos provide more information about code testing.

Objectives

Upon finishing this assignment, you should be able to:

- Solve problems using only the methods available in the `List` ADT
- Implement a reference-based (node) implementation of a stack
- Implement a stack in Java that supports generics

Submission and Grading

Attach `A4Exercises.java` and `A4Stack.java` to the BrightSpace assignment page. Remember to click **submit** afterward. You should receive a notification that your assignment was successfully submitted.

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 during grading and we will inspect your code for hard-coded solutions. [This video](#) explains stubs.

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). Plagiarism detection software will be run on all submissions.

Instructions

Part 1:

1. Download all of the .java files found in the *Assignments > Assignment 4* page on BrightSpace.
2. Part 1 uses files `List.java`, `ArrayList.java`, `Competitor.java`, and `A4Exercises.java`.
3. Read through the documentation provided in the `List.java` interface. You will be using these methods to solve problems specified in `A4Exercises.java`. The exercises work with a list of people (`Competitor.java`).
4. Compile and run `A4Tester.java`. Work through implementing each exercise one at a time. Debug the method until all of the tests pass for that method before proceeding to the next method.

Part 2:

1. For Part 2, you will complete the `A4Stack` class so that it implements the `Stack` interface. You will write a reference-based implementation using the node class provided (`A4Node.java`).
2. Read through the documentation provided in the `Stack.java` interface. There is a lot of information there that will help you set up your generic types when implementing the `Stack` interface.
3. Compile and run `A4Tester.java`. Work through implementing each stack method one at a time. Debug the method until all of the tests pass for that method before proceeding to the next method.

CRITICAL: Any compile or runtime errors will result in a **zero grade** (if the tester crashes it will not be able to award you any points for any previous tests that may have passed). Make sure to compile and run your program before submitting it!