Project 1: Grading Rubric

How will my assignment be graded?

- Grading will be divided into two portions:
    - Automatic Testing: To assess the correctness of programs.
    - Manual Inspection: For features your programs should exhibit that are not easily tested with code (such as Big-O). You CANNOT get points (even style/manual-inspection points) for code that doesn't compile or for submitting just the files given to you.
- Extra credit for early submissions:
    - 1% extra credit rewarded for every 24 hours your submission made before the due time
    - Up to 5% extra credit will be rewarded
    - Your latest submission before the due time will be used for grading and extra credit checking. You CANNOT choose which one counts.
- No credit (a 0 for the entire assignment) will be given for any of the following:
    - Non submitted assignments
    - Assignments late by more than 48 hours
    - Non compiling assignments
    - Code that violates and restrictions or “you may not” mandates (such as using array where forbidden or library imports)
    - "Hard coded" solutions
    - Code that would win an obfuscated code competition with the rest of CS310 students.
    - Non-independent work (this will also be referred to the honor court, with a sanction of “course failure”).

Earned Points for Project Completion to Specification (Manual and Automatic Testing)

For this assignment a portion of the automated testing will be based on JUnit tests and/or a manual run of your program. The JUnit tests used for grading will NOT be provided for you (you need to test your own programs!), but the tests will be based on what has been specified in the project description and the comments in the code templates. A breakdown of the point allocations is given below:

50 pts	Support Data Structure (DynamicArray – not iterator)
5 pts	Support Data Structure Iteration (DynamicArray – iterator)
10 pts	Big-O Requirements for Support Data Structure
10 pts	Simulator Basics (Simulation – not resize)
15 pts	Completion of Element, Sand, and Water
10 pts	Simulator Resizing (Simulation – resize)
+1 pt	GUI:Advanced
+5 pts	Early Submission Extra Credit

“Off the Top” Points

Note: "Off the top" points are items that will *lose* you points rather than earn you points.

Up to -5 pts	Submission folder format (see rubric below)
Up to -5 pts	Coding conventions and JavaDocs (see rubric below)
Always -5 pts	Not passing code style check (this penalty will *increase* in future projects)
Always -5 pts	Not passing JavaDoc style check (this penalty will *increase* in future projects)
Always -5pts	Compiler Warnings OR using @SuppressWarnings in non-permitted places

Rubrics

Inspection Point	High (-0pts)	Med (-2.5pts)	Low (-5pts)
Submission Format (Folder Structure)	Code is in a folder which in turn is in a zip file. Folder is correctly named.	Code is not directly in user folder, but in a sub-folder. Folder name is correct or close to correct.	Code is directly in the zip file (no folder) and/or folder name is incorrect.
Code Conventions and JavaDocs	The entire code base is well documented with meaningful comments in JavaDoc format. Occasional in-method comments used for clarity. [AND] Code has good, meaningful variable, method, and class names.	The code base has some comments, but is lacking comments on some classes/methods/ fields or the comments given are mostly "translating" the code. [AND/OR] Names are mostly meaningful, but a few are unclear or ambiguous (to a human reader)	The only documentation is what was in the template and/or documentation is missing from the code (e.g. taken out). [AND/OR] Names often have single letter identifiers and/or incorrect/meaningless identifiers. (Note: i/j/k acceptable for indexes.)