Project 3: Grading Rubric

No Credit for...

- Non submitted assignments
- Assignments late by more than 48 hours (*without exceptions*)
- Non compiling assignments
- Non-independent work
- "Hard coded" solutions
- Code that would win an obfuscated code competition with the rest of CS310 students

How will my assignment be graded?

- Grading will be divided into two portions:
  - Manual/Automatic Testing (90%): To assess the correctness of programs.
  - Manual Inspection (10%): A checklist of features your programs should exhibit. These comprise things that cannot be easily checked via unit tests such as good variable name selection, proper decomposition of a problem into multiple functions or cooperating objects, overall design elegance, and proper asymptotic complexity. These features will be checked by graders and assigned credit based on level of compliance. See the remainder of this document for more information.
- You CANNOT get points (even style/manual-inspection points) for code that doesn't compile or for submitting just the files given to you. You CAN get manual inspection points for code that (a) compiles and (b) is an "honest attempt" at the assignment but does not pass any unit tests.
- "Off the top" points refer to items that will lose you points rather than earn you points.
- 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.

Manual/Automated Testing Rubric

For this assignment a portion of the automated testing will be based on JUnit tests and 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:

Automatic Testing Breakdown:

25pts	Part 1: Linked Trees
20pts	Part 2: Parent Pointer Trees
20pts	Part 3: Linked Trees Again
15pts	Part 4: Array Trees
10pts	Part 5: Merging Trees

Automatic Testing "Off the Top" Points

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

Always -5 pts	Not passing code style check
Always -5 pts	Not passing JavaDoc style check
Always -5pts	Compiler Warnings OR using @SuppressWarnings in non-permitted places
Always -100pts	Use of "non-permitted code".

Manual Code Inspection Rubric

Inspection Point	Points	High (all points)	Med (1/2 points)	Low (no points)
Helper Code	5pts	We can locate AT LEAST four GOOD helper methods in your code.	We can locate AT LEAST one GOOD helper method in your code.	We cannot find any GOOD helper methods in your code.
Big-O Requirements	5pts	The Big-O requirements for listed methods are met and/or exceeded. Code is very efficient.	The Big-O requirements are sometimes met and sometimes not. Code could be more efficient.	The Big-O requirements are never (or almost never) met. Code is extremely inefficient.

Manual Code Inspection "Off the Top" Points

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

Up to -5pts	**Code Documentation** (see earlier projects for documentation standards)