

# Project 1: Grading Rubric

## No Credit

- Non submitted assignments
- Assignments late by more than 24 hours (*with or without tokens*)
- 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 (75%): To assess the correctness of programs.
  - Manual Inspection (25%): 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.
- 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 (no more).
  - 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:

32 pts	BetterArray
38 pts	Goblin
5 pts	Matching Output (similar tests to ones in GameTests)

**Manual Code Inspection Rubric**

"Off the top" points (i.e. items that will lose you points rather than earn you points):

- Submission format (-5pts or -2.5pts, see below)
- Code Style Basics (-5pts or -2.5pts, see below)

Inspection Point	Points	High (all points)	Med (1/2 points)	Low (no points)
Submission Format (Folder Structure)	5pts ("off the top")	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.
Big-O Requirements	10pts	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.
Code Reuse	5pts	Code is reused rather than copied.	There is some code reuse, but other code is copied and pasted which could have been reused.	There is no code reuse apparent.
Code Style Basics	5pts ("off the top")	Code passes all checks for <a href="#">cs310code.xml</a> [and] Code has a set indentation and formatting style which is kept consistent throughout and code looks "well laid out".) [and] Code has good, meaningful variable, method, and class names.	Code passes all checks for <a href="#">cs310code.xml</a> [but] Code looks "messy" and/or names often have single letter identifiers and/or incorrect/meaningless identifiers. (Note: i/j/k acceptable for indexes.)	Code does not pass <a href="#">cs310code.xml</a> checks
JavaDocs	10pts	Code passes all checks for <a href="#">cs310comments.xml</a> [and] The entire code base is well documented with meaningful comments in JavaDoc format. Each class, method, and field has a comment describing its purpose. Occasional in-method comments used for clarity.	Code passes all checks for <a href="#">cs310comments.xml</a> [and] The code base has some comments, but is lacking comments on some classes/methods/fields or the comments given are mostly "translating" the code.	Code does not passes all checks for <a href="#">cs310comments.xml</a> [and/or] The only documentation is what was in the template and/or documentation is missing from the code (e.g. taken out).