Project Code Criteria (Software or Data Engineering)

Your *source code* will be evaluated based on the following criteria.

It is a requirement that all members of the project team contribute some elements of code to the final project.

1. Effective class, method, and variable names

Names chosen for classes, methods, and variables should effectively convey the purpose and meaning of the named entity.

2. Effective top-down decomposition of algorithms

Code duplication should be avoided by factoring out common code into separate routines.

Routines should be highly cohesive. Each routine should perform a single task or a small number of highly related tasks. Routines that perform multiple tasks should call different subroutines to perform each subtask. Routines should be relatively short in most cases. [Rule of Thumb: Many routines will be less than 20 lines. Almost all routines will be less than 50 lines. Routines longer than 100 lines should be rare.]

3. Code layout should be readable and consistent

Your project must have a clean codebase.

The layout of your code should be readable and consistent. This means things like placement of comments, code indentation, wrapping of long lines, layout of parameter lists, etc.

4. Effective source tree directory structure

The source code for your project should be effectively organized into subdirectories. Something along the lines of that discussed in class would be appropriate. Old files must be removed from your project.

5. Effective file organization

Your source code should be effectively organized into multiple files. Lumping all of your code in one or two files is not acceptable.

6. Correct exception handling

Your program should handle exceptions properly.

7. Good unit test cases

Unit test cases will be evaluated according to the requirements in the project

8. API

specification.

Your project must include the use of an existing API from the internet (not applicable for game projects)

9. OOP

Your project must include the use of key OOP principles to structure your program.

10. Libraries

Your project must include the use of libraries covered in the course, such as itertools, collections or libraries you research yourself.