Variables:

- Required: Starts with lower case letter
- Required: Use descriptive names (exception for “standard loop variables” used in that context (‘i', ‘j’, ‘k’))
- Required: Name is not a verb (i.e. doesn’t “sound like” a function)
- Preferred: Use a case where non-first words start with upper case letters (eg. ‘numQuizzes’, ‘numberOfValues’, etc.)
- Required: Consistent naming scheme (use of case, underscores, etc.)

Constants:

- Required: Name is in all upper case letters
- Required: Underscores separate individual words of identifier (eg. ‘NUMBER_OF_EXAMS’)

Functions:

- Required: Starts with lower case letter
- Required: Use descriptive names
- Required: Name is a form of verb (i.e. “sounds like” functionality (i.e. ‘computeAverage’ as opposed to ‘average’)
- Preferred: Use a case where non-first words start with upper case letters
- Required: Consistent naming scheme (use of case, underscores, etc.)

Curly brace placement:

- Required: Consistent placement (i.e. always on next line, indented same amount as prev line, etc.)
- Preferred: Use curly braces to enclose code in a compound statement after an “if” or a loop, even if only one statement is currently needed

Whitespace use:

- Required: No line extends past 80 characters long, for any reason
- Required: No use of tabs in source code
- Required: Indentation with consistent number of spaces through entire source code
- Required: Every “block” or “compound statement” or “chunk of code within curly braces” is indented

Comments / Documentation:

- Required: “Header block” on every source code file – must include programmer’s name, approximate date the code was written and a brief statement about the purpose the code serves.
- Required: Inline documentation (i.e. comments) – especially to describe areas of your code that are not especially “self documenting” or “very obvious” to a reasonably knowledgeable reader of your source code

Program organization, Design:

- Required: Non-‘main’ function prototypes are specified and commented prior to the ‘main’ function.
- Required: Non-‘main’ function definitions are implemented after the ‘main’ function
- Required: Avoid use of “while (true)” type loops that use “break” to stop the loop – use proper loop conditions instead
- Required: Avoid use of “break” and “continue” to alter flow of control in loops
- Required: Use only topics covered in class – do not use libraries, functions, concepts, keywords, datatypes, etc., that have not been discussed in class as of the time the project specifications are posted

Other things to look out for:

- “Magic numbers”: essentially any literal values in your code not bound to a variable or constant (exception for “standard values” that are obvious and clear in the code)
- “Duplicated code” – same code to perform same functionality in multiple places – even minimal amounts of duplicated code is dangerous and must be avoided when at all possible
- Overly complicated code – you should avoid writing complicated or difficult to understand code when unnecessary

Disclaimer:

- While I tried to capture most of the significant style and design type issues we will be looking for when grading your programs, this is not necessarily a complete list. You may receive deductions for other style- or design-related issues that are not listed here when they are noticed in your submissions.