CS& 132 C++ Style Guide

This style guide is generally based on the coding style used in the course textbook. However, the styles applicable to indenting and function comments are more strictly defined here than in the textbook.

Naming Conventions

- Constants are ALL_CAPS nouns (i.e. EARTH_RADIUS)
- Variables are lowerCamelCase nouns (i.e. itemCost)
- Function names are lowerCamelCase verbs or actions (i.e. calculatePay)
- Struct and Class names are UpperCamelCase nouns (i.e. MovieData)

Whitespace and Indentation

- “#include” and “using” statements start in column 0.
- Each arithmetic operator has one space to either side of it.
- Commas and semi-colons have no space before them.
- Commas have exactly one space after them.
- Parentheses have no space to either side of them unless required by the adjacent element.
- Escape characters have no space to either side of them unless the space is required for output.
- Indentation is 6 spaces (this is default in VS2017 -- convert tabs to spaces in other editors).
- All code uses a style where the opening and closing braces for any statement block are on their own lines and both in the same column, with contained statements indented one tab .
- A strict indentation format is maintained throughout, with an indentation of one tab within every pair of braces, leading to nested indentation as necessary.
- Function headers start in column 0.
- A blank line separates functions.
- Multiline statements begin in the same column as any single-line statement would in the same situation, and subsequent lines of the statement are indented two tabs or in such a way as to line up in a logical fashion with the line before it, at your discretion.
- For “if-else” and iterative structures containing only one statement, braces are optional, but the statement is always on its own line and indented from the structure header.
- Switch statements have “case” headers indented one tab and all statements included in that case on separate lines and indented one more tab.
- Address-of and indirection operators (the & and the *)are attached to a data type or a variable name with no space in between.

Comments

- Every source file contains a complete identifying comment section that contains the project name, filename, authors' names, last modification date, and an overall description of what the program does.
- Comments are written with the understanding that others reading the code understand programming as well as the programmer and do not need obvious explanations.
- End-of-line comments are used sparingly to explain specific lines of code as necessary. (Proper naming convention limits the need for end-of-line comments.)
- The multiline comment format /*...*/ is never used INSIDE functions or other code blocks.
- Every function has a function comment section that contains the function name and a description of what it does.
- The function comment section may contain input, output, precondition, and postcondition descriptions as necessary to completely define the function.
    - "input" -- lists and describes the function parameters
    - "output" -- describes the function return value
    - "preconditions" -- list everything the function expects to be in place in order to run, including the declaration and initialization of reference variables and the existence of necessary files
    - "postconditions" list everything that is changed when the function runs, including alteration of reference variables, file writes, and console outputs
- Each section of code within a function has a brief description directly before it using the single-line comment format // beginning in the same column as the code directly below it.

Coding Style

- All source code compiles with standard Visual Studio 2017 settings without error or warning.
- Variables are initialized in the same statement in which they are declared if the value is known prior to runtime
- Multiple variables may be declared (and initialized) in the same statement, but each variable is listed on its own line.
- Functions are declared <u>before</u> the "main" function and defined <u>after</u> it.
- All local variables are declared at the beginning of each function, with the exception of counter variables used in iterative structures.
- Global variables are never used, but global constants are allowed.
- Break and continue statements are used sparingly in iterative programming.
- Structs, enums, and classes are defined before the main function.