Coding style for Dr. Nelson's classes

In an effort to be able to grade your assignments more effectively and to know for sure who wrote the code, the following are a list of coding standards your programs must follow.

File Header At the top of each each file you must have a comment that identifies the programmer, when the file was started, the class for which the file is written and the assignment number. This should be before any kind of code. If you have a CVS comment, this comes after the CVS comment.

Indentation How you do your indentation can make your program readable or unreadable. You should use an indentation of 2, 4 or 8 characters. Be consistent.

Long Lines Long lines should be avoided. Printing devices tend to not have as long lines as current displays. Lines should be short enough to be printed on a single line without continuations.

C/C++ .h Files When coding in C or C++, .h files must include a mechanism that keeps multiple inclusions from being a problem. Use either the #ifdef, #define, #endif method or the pragma method.

Global Variables Global variables should be limited to information that is needed by a majority of your program. Using arguments to functions is, in many cases, the better option. On the other hand, if you find your code passing the same arguments to most of your functions, that may be an indication that they should be global variables.

Variable Names Variable names should describe the contents in some way. Be reasonable about the length. Two character names may be OK when the variable contents are easily understood with two characters. For example, I would consider "index" or "ix" as valid variable names. I would not consider "ao" to be a valid name when something like "availableobjects" or "availobjs" might be better. Also, be careful in using too many characters. Very long variable names may make code more unreadable.

Variable Declarations At each variable declaration, you should include a comment that more fully describes the variable than just the name. I should not see any code that looks similar to:

```
int a, b, c=0;
char *bf[200], that;
```

They should be more like:

```
int availobjs;  // Number of available objects for use.
int bufsize;    // Number of elements in buffer.
char *buffer[200]; // Buffer for input.
```

Functions If you find yourself doing the same thing over and over again with slight modifications, try to find a way to factor out that code and turn it into a function that takes parameters that allow the slight differences.

Comments The goal of comments is for readability and to help someone reading the code understand the code. (Often the person reading the code is the person who wrote it and good comments help you remember what you were thinking when you wrote the code.) Over-commenting is a problem. With too many comments it is difficult to see the code or the comment just says in words what the code says. For example, the following comment is worthless.

```
/* Assign x the value of y plus z. */
x = y + z;
```

Under-commenting is also a problem. Code that has no comments or very few comments is also likely to be unreadable or much more difficult to read.

Comments before functions that describe the functions and their parameters are good. This especially true for parameters that are in/out or out parameters. Pre and post conditions are appropriate here but are not required. Comments before loops should give the idea of what the loop is doing. Loop invariants here are appropriate but are not required.

Attribution In the few cases where you may use code that you didn't write, make sure it is given proper attribution. Part of the attribution must be in the file header at the top of the file. Also, if your file is some of your code and some code from another source, make sure each transition is clearly marked. If you are using modified code, make sure you clearly mark what is modified code. Your attribution must include the source of the code and how to get a copy.

Be Consistent Consistency in style improves readability.

Last modified: October 8, 2009