Coding Style Guide

This coding style guide is to be followed in this programming course. Egregious violations of these instructions may negatively affect your grade.

Comments

Always begin your file with a comment block similar to the following.

```
/*
 *  Author:       Your Name
 *  Date:         Date of modification
 *  Description:  A good description of what this module does.  In three
 *                weeks or more, you or someone else may have to make
 *                modifications to this module.  There's no way you'll
 *                remember the what's and why's after writing thousands
 *                of lines and ten's or hundred's of other modules since
 *                you saw this one last.
 */
```

Format your comments as you would a paper. They must be readable. They must precede all functionally important program features they are describing. They never duplicate what the statements say -- they clarify and expound on what the following statements are doing using English and in a way that the programming language can't.

Indentation

Each level of indentation should be no less than four spaces. Use the same level of indentation throughout the file.

Variables

Declare each variable on a line by itself at the beginning of the block. All identifier names should use "camelCase" -- the first letter is lower case and each subsequent word is capitalized. If the use of the variable is not obvious by its name, then the variable must be commented with a description of its use.

White-space

Use spaces to separate parts of a statement as you would words in a sentence. Use blank lines to separate collections of statements as you would sentences into paragraphs. Comment each paragraph of statements to inform the reader what each collection of statements is accomplishing.

Statement Continuation

Don't let statements wrap or extend too long. Break the statement into another line at a logical place before 72 columns. Indent continuation lines to show how they fit with the previous line.

Parenthesize

In expressions, never trust order precedence. Instead of

```
  if (a > 1 && b != 7) and
  x = a * b + 12
```

use

```
  if ((a > 1) && (b != 7)) and
  x = (a * b) + 12
```

Control Structure

Curly braces are required with all control structures. The opening curly braces should be on the same line as the opening statement. Statements contained within the block must be indented.

```
if (logical expression) {
    /*
     *   One or more "then" statements.
     */
} else {
    /*
     *   One or more "else" statements.
     */
}
```

```
if (logical expression) {
    /*
     *   One or more "then" statements.
     */
} else if (logical expression) {
    /*
     *   One or more "then" statements.
     */
} else {
    /*
     *   One or more "else" statements.
     */
}
```

```
switch (expression)
    case 1:
        /*
         *   One or mo
         */
        break;

    case 2:
        /*
         *   One or mo
         */
        break;
}
```

```
while (logical expression) {
    /*
     *   One or more "body" statements.
     */
}
```

```
do {
    /*
     *   One or more "body" statements.
     */
} while (logical expression);
```

```
for (initialexpression; (logical expression); update expression) {
    /*
     *   One or more "body" statements.
     */
}
```