## Hi-Ho. Hi-Ho. Let's document our code! We code all day and get good pay. Hi-Ho. Hi-Ho.

Maybe it's time for me to take a break. What do you think?!

The sole purpose of documenting your code (adding comments) is to improve the readability of your program. Comments are ignored by the compiler, and do not cause the computer to perform any action when the program is executed. The compiler actually "tosses out" the comments when the object code is generated, as they provide no use to the program itself.

So, what's the big deal? Why even take the time to bother with information that is ignored by the compiler? The answer is ***because we are not computers.*** We are humans. Although C is considered a high-level language, it can still be somewhat cryptic. It is nice if programmers add some "human" language to the program, so we as humans can easily understand it.

You must realize that code that is written is not just written once and never looked at again. Code is usually reviewed by peers( or bosses! Or ***ME!),*** it is sometimes updated because of problems or enhancements. Even if your program works perfectly, lots of times your customer will want some sort of upgrade or change. For all of these reasons, code will need to be modified. Now, I know what you're thinking. (Yes, I am psychic ☺ ) Your thinking that if you write code, you will have no trouble updating it in the future, even if it is not documented well. Well. Well. Well. Are you in for a shock! Even veteran programmers, who after just a few days being away from their code, will look at the very code that they just coded a few days previous and say to themselves: "Now what the heck, pmf (pardon my French) is this code doing?!?!" (Sometimes the word "heck" is substituted with other, more harsh words, that I cannot use in this course.) And, keep in mind another possibility. You may not be the programmer assigned to update the code that you wrote! This is very common. So, when you code, realize that other programmers will be working with your code in the future. How unsettling! This is the real world folks!

So, to ensure that your code is easily readable, understandable, modifiable and maintainable, you must document your code. Now instead of wondering WHY you should document your code, I hope that you are all eagerly asking yourself HOW do I document my code. That's the spirit!

Documenting your code is actually quite easy, if it is done from the start. That is, as soon as you type in your code, add in your comments. There are 2 types of comments:

1. Block Comments.
2. In-line Comments.

# BLOCK COMMENTS

Block comments are comments that you use to describe a segment of code that follows the comment. It typically consists of several lines of information. You start a block comment with the following symbol **/*** (slash and asterisk) and end a block comment with the following symbol */ (asterisk and slash). Some good programming standards for "where to put block comments" include the following:

- At the beginning of a function, sort of as a function header (description).
- Before the "variable declaration" section.
- Before a "loop" statement (discussed in Weeks 5 and 6).
- Before an "if" statement (discussed in Weeks 8 and 9).
- Before any calculations are to be performed.
- Before any user input (prompts - discussed in Week 4).
- Before any program output.
- To the right of an "}" (end brace) to describe what block of code the end brace is ending. (This will be a -2 deduction if you do not!)

The last one is the one most often left out.  The next program in the textbook illustrates a program with some block comments. It is actually the program we just reviewed in the prior set of notes, with some comments added. This particular example shows mainly single-line block comments. However, block comments can span several lines. I will type the example here and modify the comments slightly to illustrate how block comments can span several lines if necessary.

## PROGRAM 2.6

Slightly modified - using "Block Comments"

```
/* This program adds two integer values
   and displays the results to the user. */

#include <stdio.h>
void main(void)
{

    /* Declare variables. */

    int  value1, value2, sum;

    /* Assign initial values to variables and
       computer the sum. */
    value1 = 50;
    value2 = 25;
    sum = value1 + value2;

    /* Display all values and the
       calculated sum. */

    printf ("The sum of %i and %i is %i\n", value1, value2, sum);

    getchar( );

} /* end main */
```

Naturally, I have some comments regarding block comments. The following list provides some additional programming standards:

1. Indent comments at the same level as the code that follows it. In the above code, the comment above function main starts in column 1, just like function main does. The comments within the program are indented the same number of spaces as the code in the function main's block. As of now, this is not too big of an issue, however it will become a more important issue once we discuss **loop** and **if** statements. At that time, I will provide an example to you regarding proper indentation of block comments.

2. Skip a line before and after a block comment. This provides a "roominess" in your program which adds to the readability of the code. You don't want your code to be too "squished", otherwise it is more difficult to read.

3. Do not make your block comments very "computerese". That is, they should not contain any language which appears to be code.

For example, a **good** comment would be: **/* Display the calculated sum. */**

a **not-so-good** comment would be: **/* printf the output. */**

4. Be sure your comments are descriptive, and not vague. I don't know how many programs I've received from students that contained the following 3 block comments in them:

**/* Prompt user for information */**

**/* Perform calculations */**

**/* Output results. */**

These appear to be okay, however they really are quite vague. Particularly when you realize that almost EVERY C program prompts for information, performs some calculations, and displays some results. Your block comments should be specific to your program.

In the "Programming assignment point deductions" section of the "How You'll Be Graded topic in the "Start Here" lesson, I have listed the points that will be deducted from your programming assignments if you do not follow the commenting rules.

## IN-LINE COMMENTS*

In-line comments are typically used when you have just a few words to say on a single line. They do not span multiple lines. You start an in-line comment with the following symbol: ***//*** (two slashes). Since in-line comments can only span a single line, there is no "end" symbol.

*In-line comments are not strictly ANSI (at least the old ANSI) so please use only block comments in your programs, I have adopted a -1 per // comment in the point deductions to discourage their usage.

I used to allow in-line comments, so some of my examples may still have them in there, so be careful when you cut & paste one of those programs it will not compile cleanly in all compilers.  The book uses them too!!