Mr. Giansante

Software Design: Proper Programming Style

August 2018

Proper Programming Style

Programs that are not easy to understand are bad programs even though they may give correct answers. Frequently, programs must be modified at a later date. If the person who is to modify the program cannot understand it, then they must start over and waste all the original programmer's effort. There are a number of ways to make computer programs more understandable.

Why Have Code Conventions?

Code conventions are important to programmers for a number of reasons:

80% of the lifetime cost of a piece of software goes to maintenance.

Hardly any software is maintained for its whole life by the original author.

Code conventions improve the readability of the software, allowing engineers to understand new code more quickly and thoroughly.

If you ship your source code as a product, you need to make sure it is as well packaged and clean as any other product you create.

Coding Conventions for this Class

Choose names for variables that describe the value they will be storing. For example: if you wish to have a variable store a person's total income, call the variable Total_Income rather than simply x.

Always declare variables, even when using a language that does not require you to do so.

Prompt input and output so that it is user friendly.

Use blank lines to separate chunks of codes in a logical manner.

Break up large sections of code into Procedures or Functions whenever practical.

Follow standard guidelines and conventions for windows applications. This is especially important in the design of the graphical user interface of your application.

Indent code to show structures such as loops and decision statements. Indenting is used to visually improve the layout of a program. Indenting shows which lines are subordinate to other lines. For example, all the lines which make up the body of a loop should be indented as shown below ...

Example 1 - Indenting in a loop

```
for( int a = 10; a < 20; a = a + 1 )
{
    cout << "value of a: " << a << endl;
}
```

Example 2 - Indenting in IF-THEN-ELSE

```
if (num == 5)
{
        cout<<"You guessed right.\n";
}
else
{
        cout<<"You guessed wrong.\n";
}
```

Always include a File menu.

Always include an About command in the File menu (or Help) which contains information about the program (brief description, date of creation, etc.) and the author (name, school, class, teacher's name, etc.)

Include shortcut keys as you would see in a professional application.

Include ToolTip text where necessary.

Commenting Your Code

Comments are English statements that provide valuable information about your code.

In Visual C++, comments are preceded by //.

Multi-line comments start with /* and end with */

Comments are not processed by the Visual C++ compiler, so they will not make the compiled version of your program bigger.

Source: Problem Solving (First Ed.) J. Hume, (c)1993