Professional Practice – Day 1

Headers, in-line/code block Documentation and Whitespace

In the business world programmers will often work in teams or have to demonstrate their code to their supervisors who will often enough not have any programming knowledge. So the question is how do we make our code understandable to everyone, even the untrained?

The simple answer to that is ***readability***. Our goal is to make our programs as easily readable at a glance as possible. This means well-structured and well explained.

Structure:

It is important to structure our code in a sensible manner, this means grouping together similar or sequential code. For example getting input from a user, it makes sense to group together the two commands that prompt the user and get the information together. We call these groups of similar or sequential code **code blocks**. They are to a program what a paragraph is to an essay.

Whitespace:

This is exactly what it sounds like, the white space you see in code that helps to logically break apart the program into smaller blocks of code. For example code block to calculate the area of a rectangle could be made up of many blocks within the overall block, blocks such as creating variables, retrieving data from the user, calculations and output.

Documentation:

This is the act of commenting about a piece of code or project in simple and concise English. For now we will focus on two types of documentation, the ***header*** and **code block/in-line documentation**. The problem is when we try to write in English in our programs those words are not in the programming dictionary and will therefore break the program. To fix this problem we are given a new symbol that when used tells the compiler to ignore everything after it on the line of code it is on. The symbol is the **//** sign.

Header

The header is like a signature for a file of a program. It **ALWAYS** should be the first thing to appear in every program file. It consists of 6 important pieces of information, the author of the file, file name, project name, creation date, modified date and a description of the purpose of the file. If you have a comment that takes more than 2 lines, you can use the multi-line comment technique instead of the **//**. Simply surround all comments in a **/*** and ***/**, similar to brackets.

E.g.

```
// Author: Mr.Lane
// File Name: docDemo.js
// Project Name: ProPractice
// Creation Date: Sept. 17, 2012
// Modified Date: Sept. 18, 2012
// Description: This program is built to demonstrate to a programmer how to properly document their
//                  code
```

In-line/Code Block Documentation:

This is a very brief and concise comment about what is going on in the code immediately below the comment

E.g.

```
// Output the players name in the high score board
System.out.println("Congratulations " + username + " your final score was " + score);
```