

Writing Maintainable Programs

The Importance of Good Programming Style

- It is easier to check the program does what it's meant to, because the intention is spelt out in the comments alongside the code.
- Programmers working in a team will need to be able to understand each other's code, this will be easier if it is written following standard rules, and is commented on fully.
- If another programmer needs to take over or modify the code, this too will be easier for them if it is well written.
- Quicker to refamiliarise yourself if the code needs to be revisited in the future.

Variables

Selecting Identifiers –

An identifier is the name given to a variable or constant.

- They should have meaningful names, that communicate their purpose.
- Spaces are not allowed, so words should use underscore_ or camelCase.
- Prefixes should represent the data type.
- Reserved words cannot be used in identifiers.

Scope

- When declaring a variable the programmer needs to consider the scope.
- A local variable is declared and used within a module or subroutine, it is only available within that module. Created when that subroutine is called.
- A global variable is declared at the beginning of the code, and available throughout the code and all the subroutines.

Initialising Variables –

- This is when the variables are given a starting value, usually 0, FALSE or "".
- This gets rid of anything else which may already be stored in that part of the memory.

Using Constants –

- Constants are very useful in making code easier to read and maintain.
- They are better than using literals, and make the code more understandable.

Code Layout

- The way code is written contributes significantly to its maintainability.
- It should be organised into modules, using indentation to show control structures, and white space to format the code into logical groups of statements.

Modularised Code

- Code should be written as a series of small routines, which can be quickly seen as performing a particular one function.
- Modularisation is easier to achieve if the program has been designed using top-down techniques.

Indentation

- Every time a code structure is used that has a beginning and ending statement on separate lines, the code within the structure should be given a new level of indentation.
- This will make it clear where each structure starts and finishes, and which structures are nested within others.

Formatting

- The code should be grouped into logical blocks that perform part of the main task.
- Blank lines should be entered between them.
- This is the equivalent to paragraphs.

Commenting

- Comments are read by the programmer, but ignored by the computer.
- Should be written while the code is being written.
- There should be comments, at the program header, at each subroutine, when variables and constants are declared and the code it's self should have comments.