Good Programming Practice – Commenting Code

Comments

General references on comments:

1. Comment (computer programming) From Wikipedia
2. Code Complete, Second Edition Chapter 32. Self-Documenting Code
3. Clean Code: A Handbook of Agile Software Craftsmanship Chapter 4. Comments

SAS general references on comments:

1. SAS® Comments – How Straightforward Are They?

Summary Points About Comments

Internal Documentation.

- Comments in the code are one part of the broader category of program documentation that includes internal and external documentation. Requirement specifications and design documents are external application documentation that should be linked to internal comments.

Are Comments Good?

- Comments are not necessarily a Good Thing. Robert Martin (3) says that comments are at best a necessary evil. Comments that simply repeat what the code is doing can become misleading if they are not accurately maintained and can violate the 'Once and Only Once' good programming principle.

Let the Code Do the Talking.

- Clearly written code is the best form of internal program documentation. Good comments explain the intent of the code or give a summary of its intent -- things that cannot be stated by the code itself.

Ease of Use.

Use a commenting style that is easy to maintain. Columns of characters as decorative boarders only require additional editing when a change is made. Indent and line up comments with the code they describe to maintain the code layout.

SAS Syntax and Practice.

- SAS syntax provides several styles of comments that can be used several ways. Block comments, /*...*/, can be useful at the lowest level of detail since they can be inserted within SAS statements. Extra care should be taken with comments used within SAS macros or with macro statements. The macro comment, %*, is safest in commenting macros.

Automated Documentation.

- Several processes use specially formatted code comments as input for automatically generated summary documents. These systems can be helpful in comparing actual program functionality with design specifications.

-- Paul ok01 17:42, 27 June 2010 (UTC)

Commenting code for Clinical Trial and Healthcare related programming

General rules for commenting suggest that commenting should say why you are doing something and should not simply describe what the code is doing.

I think that the following are good reasons for using comments in Clinical programming.

1. Comment why you are doing a particular step

Relate to a requirement in the specification or analysis plan. Related to a peculiarity of the data. Clinical data is complicated and often the code makes little sense without an understanding of the data. If it is unusual then it is definitely worth saying so. Simply coding for missing values may be self explanatory but more complicated scenarios may not be and it will be a reminder at a later date as to why you coded this way and indicate how and why the code could be updated. Complex or unusual coding is worth commenting. If you are writing code that you consider is well written, but not everyone will understand, you need to consider whether to rewrite or comment. Proc SQL , macro and use of the DOW loop often fall into this category. For example it is worth saying that your intention in a DOW loop is to retain a variable or generate an array as this will help another programmer to maintain or update the code.

Examples of good and bad self documenting code