

I'm substantially changing my Matlab documentation requirements in light of my discovery of the *m2html* tool for automatic generation of Matlab documentation.

## **Part 1 – External (Memo)**

Full size, professional programs generally require at least three documents. These are the *Program Design Document*, the *User's Manual* and the *Technical Reference Manual*. In this course, you will combine these documents into a single cover memo, since your programs will be relatively short and simple. Address your memos to me as your "Engineering Manager at Acme Products." Submit your memo and source code via e-mail. Your memo should generally be no more than 3 pages long and contain the following short sections (section headings are required):

### **Introduction**

In this section, describes what the program does. The introduction should essentially restate the assigned problem and layout specifications for the work to be done.

### **Operating Instructions**

In this section, explain how a user can use your program. This section serves as the User's Manual.

### **Input and Output Description**

In this section, describe the input required by the program (if any) and the output it produces. This section should contain a clear statement regarding whether the program operates in batch or interactive mode. If the program is interactive, this section should indicate whether the user interface is graphical or text/console based. It should include descriptions of all values that must be entered during program execution, the format of any input and output data files (not your source files) and the units of all values having units.

### **Sample Solutions and Test Results**

These are solutions to the problem using some method other than the computer program. They are often done by hand, but could include spreadsheet, experimental or other results. Sometimes they are solutions for some special cases of the problem that are particularly easy (or at least possible) to solve by hand. These results should be **explicitly compared** to those from the program to demonstrate that the program works. This material is often best presented in tabular format. The number of specific cases done should be sufficient to prove beyond a reasonable doubt that the program works.

## Dependencies and Design Details

Dependency and other program design details should now be included in comments at the start of each module (script or function). Scripts should start with a single comment line stating the fundamental purpose or use of the script. Functions should include this comment as the line immediately following the *function* statement (typically the first and second lines in the file). These are commonly called *lookfor* or *H1* lines.

These initial comment lines should be followed by a block of comments not including any blank lines. This block should provide a concise description of what the script or function does and how it does it. A list of pre-conditions (the existence of input data files, the presence of a user (for interactive program), etc.) and post-conditions (the generation of figures, the opening of files and the corresponding validity of file IDs, etc.), for functions a list of incoming or input arguments and/or returned quantities. It should also include a list of exceptions the script or function can throw. This block of comments should end with creator, revision, version and copyright information. The overall impact of these comments should provide the reader with sufficient information to understand, use and modify the script or function with minimal additional analysis.

## Appendices (as appropriate)

Include with main document and source files combined in a single archive (.zip, 7z, etc.) file.

### Source Code Listing

The complete source code (.m) files. Remember to include plenty of comments (see below) in your source code.

### Screen Shots (if appropriate)

Screen shots of the program in operation with explanatory text. These are particularly important for GUI programs but may be useful for console mode programs. Screen output captured from your PC (use Fn PrtSc on most laptops).

### Data Listings (if any)

Input and output data files as appropriate.

### Literature Sources (if any)

Citations (references) to any literature used in the design and development of the program. This could include class handouts, websites (like the Medical Algorithms site, [www.medal.org](http://www.medal.org)), textbooks, professional journal articles, etc.

## Part 2 – Internal (header)

See the "Dependencies and Design Details" section above.

## Part 3 – Internal (general purpose)

Other Comments – Your program must contain enough comments to make its operation easy to understand. This doesn't mean every line should be commented, just each major section or block of code. Comments generally should focus on what is being done, not how it is being done (that should be obvious from your code). The exception to this rule is cases in which you're doing something "tricky" or obscure, in which case a detailed explanation should be provided. Note that in general tricky or obscure code should be avoided (that is, it should only be used when necessary for performance or other compelling reasons).

Other Internal Documentation – A logical program design, consistent indentation and mnemonic variable, class and function names go a long way toward creating a well documented a program and these aspects will be considered in the grading.