Guidelines for Commenting Programs

☚ Back to Labs

Commenting is meant to make your program understandable to someone else or even to yourself six months later. It can even be vital in debugging your own code. So don't get lazy – comment as you go, not after you're done.

In MATLAB, the comment character is %. Anything following a % will be ignored by MATLAB as a comment.

Block comments

The internal block comment is meant to give an overview of the construction and operation of the routine. If the reader wants more detail, he or she can move on to read the code, which will have its own comments about the code itself.

Block comments are put at the beginning of the program/function. Code would follow directly underneath.

An example of a bad block comment:

```
% Function FIND_BIGGEST_QUAKE
%
% Purpose: Find the biggest earthquake.
%
% Parameters:
% quake_list (IN) -- the quakes
%
% Returns: The biggest quake
%-----------------------------------------------------------------
```

What's wrong with this comment?

- The description of the purpose of the function is useless; we could have figured as much from the name of the function alone!
- The structure of the quake_list parameter is not even mentioned.
- How is the list size measured? By number of items or by index position of the last item?
- Exactly what about the biggest earthquake is being returned? Its position in the list? Its measured size? The entire entry from the list?

Compare the first attempt to this one:

```
 % Function FIND_BIGGEST_QUAKE
%
% Purpose: Identify the earthquake from the list of earthquake
% intensities (quake_list) that has the largest magnitude.
% It is assumed that the given list of quake intensities is
% an array-based list of unordered Richter Scale
% magnitudes; this function performs a simple sequential
% search through the array to locate the position of the
% largest magnitude (the largest value) in the list.
%
% Parameters:
% quake_list (IN) -- the array of earthquake magnitudes. This
% is just an array of real numbers; the first magnitude
% is assumed to be at index 0.
% num_entries (IN) -- the quantity of magnitudes in quake_list.
%
% Returns: The index (position) of the largest earthquake
% magnitude in the quake_list array.
%-----------------------------------------------------------------
```

Inline Comments

Inline comments explain the detailed workings of your code. They are especially important for identifying variables. They also delineate functional sections of the code and clarify the reasons for each major operation.

For example:

```
% tank3.m - script file to simulate water tank
%
% Linearly interpolated nonlinear area term
%
% This version demonstrates how proper state logic coding
% can eliminate errors such as 'h' falling too low after
% emptying as we observed in tank.m.
%
% set h (height) to zero (no water), and state to zero
% meaning that tank starts empty and filling commences.
%

h=0;
state=0;

% Initialize other variables and constants
%

t=0;         % time (seconds)
g=9.81;      % m.sec^-2
Cd=0.6;      % discharge coefficient of bottom hole
Ah=pi*(0.04^2 - 0.02^2); % water flows out through annular orifice
                         % 4 cm OD and 2 cm ID.
Htop=0.3;    % top level of tank
Hv=0.2;      % level at which valve starts to close
Qimax=0.01/60; % 10 litres per minute (m^3/sec)
```

Note how in this code fragment there is very little doubt about why each statement is there. Things are explained in the context of the problem.

The keys of inline commenting:

- USE WHITESPACE for legibility
- Label units and variables
- Try to use variable names which are descriptive. DO NOT use x, a, z for everything.
- Label why and how, not just what.