CMPSC 200 Programming for Engineers with MATLAB
Programming Style Convention

Many different organizations maintain standards for code documentation – the purpose of using a style convention in this class is twofold: 1) to get you use to documenting your code to a given standard, and 2) to facilitate grading a large number of programs.

Note: The easiest way to line up lines is to tab over with the tab key.

Comments at the top of your program

Pick your favorite format between Option 1 and 2 (both are acceptable):

Option 1:

```
% Programmer:  Your Name
% Section:     1 (enter your correct section number)
% Lab/Project: 1 (enter the correct type and number)
% Date:        September 3, 2014 (update with correct date)
% Description: A description of what the program does; limit this to just
%              the highlights - only a few sentences
```

Option 2:

```
%{
    Programmer:  Your Name
    Section:     1 (enter your correct section number)
    Lab/Project: 1 (enter the correct type and number)
    Date:        September 3, 2014 (update with correct date)
    Description: A description of what the program does; limit this to just
                 the highlights - only a few sentences.
%}
```

Variables and Constants

Each variable or constant should be allocated one per line, so don't allocate multiple variables on one line of code. When you declare a constant or a variable, include a comment with the variable or constant's units and what it represents; include empty braces for variables that are dimensionless. For example:

```
g = 9.81;           % [m/s^2] gravitational acceleration at sea level
mu_static = 0.5;    % [ ] coefficient of static friction
```

Constants should be allocated at the top of a program, underneath the comments at the top of the program and the `clc` and `clear` statements. Listing them together forms a "data dictionary" of constants that can be easily seen by the programmer and anyone that later has to view your code.

Subplots

So that it is clear what commands belong to each subplot, indent the commands with the tab key.

```
subplot(1,2,1)
    plot(w,x)
    title('My title #1')
subplot(1,2,2)
    plot(y,z)
    title('My title #2')
```

Function Files

It is important for functions to know what the function does, what are its inputs, and what are its outputs. Please format your function files like this:

```
% Programmer:  Your Name
% Section:     1 (enter your correct section number)
% Lab/Project: 1 (enter the correct type and number)
% Date:        September 3, 2014 (update with correct date)

function [ output_args ] = myfunction( input_args )
%   Inputs:     1st input - comment as described in "Variables and Constants"
%               2nd output - comment same as above
%   Outputs:    1st output - comment same as above
%               2nd output - comment same as above
%   Processing: A description of what the function does; limit this to just
%               the highlights - only a few sentences
```

(some calculations are performed underneath the comments, documented as usual)

```
end
```

Include as many inputs and outputs as you have, each on a separate line. The listed variables `output_args` and `input_args` are the MATLAB defaults when you open a new function. Change these so that they are your variable names (as appropriate).

Selection Structures

To increase readability, format your selection structures with indents. You may include parenthesis for your conditions if you feel they make your code more readable. For example:

```
if something
    calculation(s)
elseif something_else
    calculation(s)
else
    calculation(s)
end
```

Similar formatting is requested for switches, for example:

```
switch x
    case 1
        calculation(s)
    case 2
        calculation(s)
    otherwise
        calculation(s)
end
```

Repetition Structures

To increase readability, please format your repetition structures with indents. You may wish to include parenthesis for your `while` loop conditions if you feel they make your code more readable. An example for `for` loops is:

```
for i = matrix
    statment(s)
end
```

Similarly, for a `while` loop:

```
while condition
    statement(s)
end
```

Nesting

If you have multiple selection structure or repetition structures nested within each other, please continue indenting. Comment next to each `end` stating at least what statement selection or repetition structure, though you may be more detailed than that if you wish. For example:

```
for i = 1:10
    while condition
        switch x
            case 1
                if y < 0
                    statement(s)
                else
                    statement(s)
                end % end if
            case 2
                if y < 0
                    statement(s)
                end % end if
        end % switch
    end % while loop
end % for loop
```

Comments and Style

Comment your code, particularly if you do anything "clever." What may be obvious to you now may not be obvious to a colleague, or even to yourself a week later. Explain calculations where appropriate; this could include equation numbers if you use them from a given document. Don't comment just for the sake of commenting. Comments shouldn't just decode MATLAB syntax into English. You may assume the reader is at least minimally proficient in the language.

Please make sure that your code and comments are free from spelling and grammar errors – pay particular attention to spelling variable names correctly. MATLAB will assume you are trying to allocate a different variable if you don't type a variable exactly the same each time (spelling, capitalization, *etc*.).