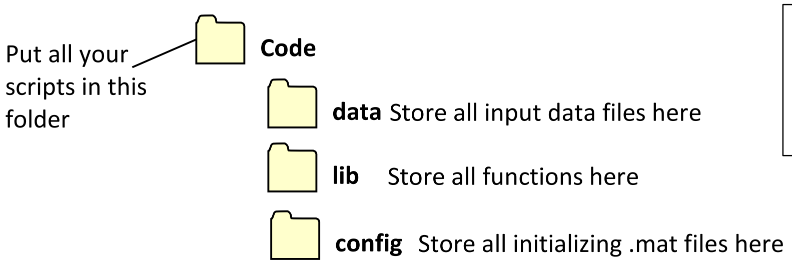
Matlab Coding Standards

Updated: 2016 MAY-04

1. Organize application scripts and data files into the following folder structure and load and save data accordingly:



Put all your scripts in this folder

Code

data Store all input data files here

lib Store all functions here

config Store all initializing .mat files here

You can adjust this structure to accommodate assignment and project folders, but maintain the general data flow

2. Include a Comment Header in all your application scripts and functions. The comment header should have the following structure:

```
%--scriptname (application scripts should be followed by .m, function with ())
%
% Author: D MacIsaac
% Id:    8510804
% Date:  2006-08-01 (if you update the script, add the update date here)
%
% Description:  A short description of what the script does
```

3. Clear the command window and all variables at the start of each script whenever possible.
4. Block program commands, include a descriptive title as a comment for each block and separate blocks with 1 blank space.
5. Include in-line comments to clarify cryptic commands, or commands using assumptions and/or analytic derivations
6. User descriptive variable names
7. Consider in-line commenting all variables with their units
8. Use a ‘Setting Parameters’ block whenever possible.
9. Use a ‘Displaying Results’ block whenever possible.
10. Use a ‘Loading data’ block when appropriate, and always exit gracefully from the application when data is not found
11. Use program structures and tabbed indentation to indicate commands within a structure

```
if…elseif…else…end
while…end
for…end
switch…case 1…case 2…case…n…otherwise…end
```

12. Create functions for repetitive code or blocks which have potential for reuse
13. Consider using nested functions to provide cleaner, easier to read code
14. Use a separate command to set return arguments in a function
15. Use data structures to collect related data where appropriate
16. Remove all dead code where possible
17. Use a model-view-control design strategy when building application with GUIs
18. Program to specification while optimizing capacity to generalize, and balancing readability with efficiency