# OHD General Programming Standards and Guidelines Peer Review Checklist

| Reviewer's Name: | Cameron Carsley | Peer Review Date: | 3/13/2018 |
|---|---|---|---|
| Project Name: | | Project ID:<br><br>Enter if applicable | |
| Developer's Name: | Taylor Dudunake | Project Lead: | |
| Review Files & Source code | | | |
| | | | |
| | | | |
| | | | |
| Code Approved | | | |

This checklist is to be used to assess source code during a peer review.  Items which represent the code being reviewed should be checked off.

Refer to the *OHD General Programming Standards and Guidelines* document for more complete descriptions and examples of the items listed below.

# 1.  Internal Documentation

__N/A__ Comment block exists at the beginning of the source file containing at least the following information: original author's name, file creation date, development group, and a brief statement of the purpose of the software in the file.

__x__ Each subroutine or function in the file is preceded by a comment block which provides the following information:  routine name, original author's name, routine's creation date,  purpose of the routine,  a list of the calling arguments

(their types and what they do), a list of required files and/or database tables, the routine's return value, error codes and exceptions processed by the routine, and a modification history indicating when and by whom changes were made.

# 2.  Programming Standards

__x_  Consistent indentation of at least 3 spaces is used to distinguish conditional or control blocks of code. TABS NOT USED FOR INDENTATION.

__x_  Inline comments are frequently used and adequately describe the code.

__x__  Structured programming techniques are adhered to.

__x__  Subroutines, functions, and methods are reasonably sized.

__x__  The routines in each source file shall have a common purpose or have interrelated functionality.  Methods in a class support its functionality.

__x__  The name of the file, script or class represents its function.

__x__  Function and method names contain a verb, that is, they indicate an action.

__x__  Meaningful variable names are used.

__-__  Variables are initialized prior to use.

__x__  Braces are used consistently

__x__  There are no compiler warnings

# 3.  Programming Guidelines

__x__  Source file line lengths are 80 characters or less.

__x__  Spacing is used correctly to enhance the source code's readability.

_n/a_  When continuing lines of code on new lines, they are broken after a comma or an operator. Higher level breaks are used instead of lower level breaks.

_n/a_  Wrapped lines of code are aligned with the beginning of the expression at the same level on the previous line.

_n/a_  Multiple line variable declarations are preceded by a type.

__x__  Program statements are limited to one per line.

__x__  Nested program statements are avoided.

\_\_x\_\_ Parentheses are used to remove operator precedence ambiguity and to improve code readability.

\_\_x\_\_ Inline comments constitute approximately 20% of the total lines of code in the program, excluding the file and routine documentation blocks.

\_\_x\_\_ The software reflects a balance of coding for efficiency and coding for readability.

\_\_-\_\_ Meaningful error messages are used.

\_n/a\_ System calls which acquire system resources are tested for error returns.

\_\_x\_\_ Routines and methods are contain no more than 200 executable lines.

\_\_x\_\_ The number of routines in a source file is kept to a minimum for procedural languages.

# Reviewer's Comments:

Taylor, your autoregressive models look like they're doing what they are supposed to do.  Your use of markdown cells with varying sizes of text make the document very easy to read and to follow along with exactly what is going on in each step.  One thing I noticed is that comments are fairly few and far between in the actual code, which can make it hard to pick back up after a break of looking at it.  Overall, great job!

Cameron