**ESRI ArcPro**

**ORMAP Tools Committee**

**Programming Standard**

Programming standards will apply to production programs/scripts (ie. Python) and “low code” program environments (ie. Arcade) that will be shared with other members of the Tools Committee.

**Program Header**

* Identify program name
* Purpose (Short description)
* Dependencies (if applicable)
* Initial State (if applicable)
* Post State (if applicable)
* Classes/Functions/Routines (if applicable)
* Primary Steps
* Author
* Date
* Updated By (if applicable)
* Updated Date: (if applicable

Example:

##############################################################

# ZoomTo.py

#

# Purpose: Zooms to a map index or tax lot. This script is intend to run from a  
# ArcGIS Pro Tool box.  
#

# Dependencies: ArcGIS Pro Toolbox. An active map with the tax lot and map index  
# feature classes. Requires the user to input a valid map index value. The Tax lot value is optional.

#  
# Initial State: Active map has map index and tax lot layer.

#

# Post State: Map is centered on selected features.

#

# Primary Steps: finds the map index layer and tax lot layer in the map. Selects the target features using the

# values entered in the geoprocessing form.

#  
# Author: O. R. Map  
# Date: 1/1/2021

# Updated By: D. Polk

# Date: 2/1/2021

# Update Description: Changed the hard coded to Map Index to a script parameter. The change is on line 15.

# Added the variable mapIndexLayer.  
##############################################################

**Program Body**

* Identify major steps
* Code should be uniform with readable indentations
* Identify Variables (Camel Case)
  + quarter
  + quarterQuarter
  + townshipDirection
  + mapNumber
* Identify any unique/unusual code
* Clearly identify any areas where a non-developer will need to interact with the code  
   (# --- County Specific Code Follows ---- #)
* Identifying Definitions (Def) using a name distinguishing as a function or method. Document the inputs, returned value, or methods completed action.
  + Function returning a map tax value example.

#Calculates the map and tax lot vaue. Inputs MapNumber and Taxlot. Returns MapTaxlot

functionCalcMapTaxLot(mapNumber, taxLot)”

* + Method appending features to another feature class example

#Appends taxlots to an existing tax lot feature class. Inputs Selected taxlots, target feature class.

methodAppendTaxlots(inputTaxLots, targetTaxLotLayer):