SAS Good Programming Checklist

# Generally accepted standards

Reference: <https://advance.phuse.global/display/WEL/Good+Programming+Practice+Guidance>

* Make backup copy of program before updates
* Use program documentation
  + E.g. use headers and comments
  + Write documentation in English
  + Inbed comment with the body of the program code should detail modular flow
* Write only one statement per line
* Use indentation to arrange the code clearly
* Use unique names for dataset and files within the program/macro
* Reference datasets and files explicitly in each data step and procedure
* Reset global options to original settings if changed in the program/macro
* Delete temporary dataset after program execution
* Use run/quit at the end of each DATA step/PROC statement
* Use defensive coding
* Optimize the data, do not re-read data
* Do not hardcode data
* Do not manually edit output
* Do not overwrite input data
* Do not use SAS keywords for dataset and variable names
* Always include a final ELSE/OTHERWISE clause in SELECT/IF/CASE statement
* Read all external data at the top of the program
* Use lowercase for all programming
* Group similar statements together
* Define new variables with the ATTRIB statement in order to ensure the variable properties are correct instead of implicit determined by the data/circumstances
* Use logical groups to separate code into blocks
* Double space between sections
* Perform one task per module or macro
* When converting variables to numeric and vice-versa, use the PUT and INPUT functions to explicitly convert the variables to ensure that it is done in the way intended
* Keep naming conventions consistent across dataset and programs
* Use self-explanatory names for variables and datasets
* Insert parentheses in meaningful places to clarify sequence of mathematical or logical operations
* For DO loops, place the END statement in the same positions as the DO statement so that they can be easily matched
* Do not use tab for indentation
* Split DATA steps into logical parts
* Left justify global statements and DATA and PROC statements and their corresponding RUN and QUIT statements
* Separate DATA steps and PROC statements with at least one blank line
* Precede each DATA step and PROC statement with a comment

# Additional standards for myself

* Use PROC FORMAT for 1:1 re-categorization of data
* Import/create at the top of the program
* Import/create formats at the top of the program
* Use a standard header for every program
* Put period at the end of macro variables