Macros in SAS Software

Duration: 2 units ***CEUs: 1.2***

AUDIENCE

This course is designed for the experienced SAS user. You will learn the theory and methods of SAS Macro programming and how to apply them to common problems. Creating simpler and more efficient SAS code is addressed. Quite often programs are re-run with minor changes. This promotes multiple copies of the same code. We show how to avoid copying programs used many times for processing and how to create a generic version of the code for future processing. Large programs can be reduced to single lines. Managing of processes is much easier. We discuss the creation of routines in SAS. Other people can use routines created, using macros without having to re-code from scratch.

BENEFITS

Students will understand how macros in SAS software will help them develop programs more efficiently. Students will:

- *Understand why we use macros*
- *Be able to write and debug macro code*
- *Develop macros that dynamically build program code*
- *Create and use macro variables*

PREREQUISITES

This course requires prior SAS programming knowledge, including familiarity with the following:

- *Creating and running SAS programs*
- *Applying the LIBNAME, FILENAME, TITLE and OPTIONS statements*
- *Using the DATA step to access a SAS data set or an external data file*
- *Incorporating DATA step programming statements, such as, IF-THEN/ ELSE, DO WHILE, DO UNTIL and the DO LOOP.*
- *Applying character functions, like SUBSTR, SCAN, INDEX, and UPCASE*
- *Specifying the LENGTH and RETAIN statements*
- *Applying SAS data set options, including DROP=, KEEP=, and OBS=*
- *Using the WHERE clause to subset data*
- *Understanding and applying SAS date values, including date constants*
- *Using base SAS procedures, including SORT, PRINT, CONTENTS, MEANS, FREQ, TABULATE, and CHART*

This knowledge base may be acquired by taking Programming II: Data Manipulation Using the Data Step.

COURSE TOPICS

Introduction to Macros

- *How the SAS macro language works*
- *Introduction to tokenizing, compiling and executing a SAS program*
- *How the macro processor works*

Applying Macro Variables in a SAS Program

- *Applying automatic macro variables*
- *Designing customized macro variables*
- *Substituting the macro variable value in strategic points in the program code*
- *Incorporating macro variables in SAS programming*
- *Displaying macro variable values in the SAS log*
- *Applying quoting functions with macros*

Incorporating SAS Macros in the Data Step

- *Designing macro variables during DATA step execution*
- *Indirectly referencing macro variables*
- *Resolving macro variables during DATA step execution*
- *Understanding the functionality and application of the SYMGET function and SYMPUT routine*
- *Using the INTO clause to build macro variables during PROC SQL execution*

Running Macro Programs in SAS Software

- *Designing and implementing simple macros*
- *Specifying conditional coding inside a macro*
- *The macro compilation and execution processes in the macro processor*
- *SAS system options used for debugging macros*
- *Reviewing error and warning log messages dis played by the macro processor*
- *Designing and using macros containing parameters within them*
- *Using positional and keyword parameters in macro calls*
- *Differences between global and local symbol tables*
- *Nested macros and symbol table hierarchies*

Techniques for Storing Macros

- *Understanding the autocall feature*
- *Permanently storing and using compiled macros*
- *Writing efficient macro programs*

Software Used: Base SAS Software