Base SAS Programming

1. SAS Programming 1: Essentials

Duration: 2.5 Days

This course is for users who want to learn how to write SAS programs to access, explore, prepare, and analyze data. It is the entry point to learning SAS programming for data science, machine learning, and artificial intelligence. It is a prerequisite to many other SAS courses.

Learn how to

use SAS Studio and SAS Enterprise Guide to write and submit SAS programs
access SAS, Microsoft Excel, and text data
explore and validate data
prepare data by subsetting rows and computing new columns
analyze and report on data
export data and results to Excel, PDF, and other formats
use SQL in SAS to query and join tables.

Who should attend
Anyone starting to write SAS programs

Prerequisites
Before attending this course, you should have experience using computer software. Specifically, you should be able to

- understand file structures and system commands on your operating systems
- access data files on your operating systems.

No prior SAS experience is needed.

Course Contents

Essentials

- the SAS programming process
- using SAS programming tools
- understanding SAS syntax

Accessing Data

- understanding SAS data
- accessing data through libraries

- importing data into SAS

Exploring and Validating Data

- exploring data
- filtering rows
- formatting columns
- sorting data and removing duplicates

Preparing Data

- reading and filtering data
- computing new columns
- conditional processing

Analyzing and Reporting on Data

- enhancing reports with titles, footnotes, and labels
- creating frequency reports
- creating summary statistics reports

Exporting Results

- exporting data
- exporting reports

Using SQL in SAS

- using Structured Query Language in SAS
- joining tables using SQL in SAS

2. SAS Programming 2: Data Manipulation Techniques

Duration: 2.5 Days

This course is for those who need to learn data manipulation techniques using the SAS DATA step and procedures to access, transform, and summarize data. The course builds on the concepts that are presented in the *SAS Programming 1: Essentials* course and is not recommended for beginning SAS software users.

Learn how to

understand and control DATA step processing
create an accumulating column and process data in groups

manipulate data with functions
convert column type
create custom formats
concatenate and merge tables
process repetitive code
restructure tables.

Who should attend
Business analysts and SAS programmers

Prerequisites
Before attending this course, you should be able to do the following:

- write DATA step code to subset rows and columns, compute new columns, and process data conditionally
- sort tables using the SORT procedure
- apply SAS formats

Course Contents

Controlling DATA Step Processing

- setting up for this course
- understanding DATA step processing
- directing DATA step output

Summarizing Data

- creating an accumulating column
- processing data in groups

Manipulating Data with Functions

- understanding SAS functions and CALL routines
- using numeric and date functions
- using character functions
- using special functions to convert column type

Creating Custom Formats

- creating and using custom formats
- creating custom formats from tables

Combining Tables

- concatenating tables

merging tables
identifying matching and nonmatching rows

Processing Repetitive Code

using iterative DO loops
using conditional DO loops

Restructuring Tables

restructuring data with the DATA step
restructuring data with the TRANSPOSE procedure

SAS Macro Language 1: Essentials

Duration: 14.0 hours

This course focuses on the components of the SAS macro facility and how to design, write, and debug macro systems. Emphasis is placed on understanding how programs with macro code are processed.

Learn how to

perform text substitution in SAS code
automate and customize the production of SAS code
conditionally or iteratively construct SAS code
use macro variables and macro functions.

Who should attend: Experienced SAS programmers who have a sound understanding of DATA step processing and who want to write SAS programs that are reusable and dynamic

Prerequisites

Before attending this course, you should have completed the *SAS Programming 2: Data Manipulation Techniques* course or have equivalent knowledge. Specifically, you should be able to

use a DATA step to read from or write to a SAS data set or external file
use DATA step programming statements such as IF-THEN/ELSE, DO WHILE, DO UNTIL, and iterative DO
use SAS data set options such as DROP=, KEEP=, and OBS=
use character functions such as SUBSTR, SCAN, INDEX, and UPCASE
form subsets of data using the WHERE clause
create and use SAS date values and constants
use SAS procedures such as SORT, PRINT, CONTENTS, MEANS, FREQ, TABULATE, and CHART.

Software Addressed

This course addresses Base SAS software. This course is appropriate for students who are using SAS 9 software.

Course Contents

Introduction

overview of SAS Foundation
course logistics
course data files

purpose of the macro facility
program flow

Macro Variables

introduction to macro variables
automatic macro variables
macro variable references
user-defined macro variables
delimiting macro variable references
macro functions

Macro Definitions

defining and calling a macro
macro parameters

DATA Step and SQL Interfaces

creating macro variables in the DATA step
indirect references to macro variables
creating macro variables in SQL

Macro Programs

conditional processing
parameter validation
iterative processing
global and local symbol tables

Learning More

SAS resources
beyond this course

Supplemental Materials

program flow

SAS SQL 1: Essentials

Duration: 17.5 hours

This outline is provisional and subject to change.

This course teaches you how to process SAS data using Structured Query Language (SQL).

Learn how to

- query and subset data
- summarize and present data
- combine tables, including complex joins and merges
- create and modify table views and indexes
- replace multiple DATA and PROC steps with one SQL query.

Who should attend: SAS programmers and business analysts

Prerequisites

Before attending this class, you should be able to

- submit SAS programs on your operating system
- create and access SAS data sets
- use arithmetic, comparison, and logical operators
- invoke SAS procedures.

You can gain this experience from the *SAS Programming 1: Essentials* course. No knowledge of SQL is necessary.

Software Addressed

This course addresses Base SAS software. This course is appropriate for students who are using SAS 9 software.

Course Contents

Introduction

- overview of SAS Foundation
- course logistics
- course data files
- introducing the Structured Query Language

Basic Queries

overview of the SQL procedure
specifying columns
specifying rows

Displaying Query Results

presenting data
summarizing data

SQL Joins

introduction to SQL joins
inner joins
outer joins
complex SQL joins

Subqueries

noncorrelated subqueries
in-line views

Set Operators

introduction to set operators
the UNION operator
the OUTER UNION operator
the EXCEPT operator
the INTERSECT operator

Creating Tables and Views

creating tables with the SQL procedure
creating views with the SQL procedure

Advanced PROC SQL Features

dictionary tables and views
using SQL procedure options
interfacing PROC SQL with the macro language

Learning More

SAS resources
beyond this course

SAS Programming 3: Advanced Techniques and Efficiencies

Duration: 21.0 hours

This course is for SAS programmers who prepare data for analysis. The comparisons of manipulation techniques and resource cost benefits are designed to help programmers choose the most appropriate technique for their data situation.

Learn how to compare various SAS programming techniques that enable you to

- benchmark computer resource usage
- control memory, I/O, and CPU resources
- create and use indexes
- combine data horizontally
- use hash and hiter DATA step component objects and arrays as lookup tables
- compress SAS data sets
- sample your SAS data sets
- create and use SAS data views
- safely reduce the length of numeric variables
- create user-defined functions and informats.

Who should attend: Experienced SAS programmers

Prerequisites

This course is **not** appropriate for beginning SAS software users. Before attending this course, you should have at least nine months of SAS programming experience and should have completed the *SAS Programming 2: Data Manipulation Techniques* course. Specifically, you should be able to do the following:

- understand your operating system file structures and perform basic operating system tasks
- understand programming logic concepts
- understand the compilation and execution processes of the DATA step
- use different varieties of input to create SAS data sets from external files
- use SAS software to access SAS libraries
- create and use SAS date values
- read, concatenate, merge, match-merge, and interleave SAS data sets
- use the DROP=, KEEP=, and RENAME= data set options
- create multiple output data sets
- use one-dimensional array processing and DO loops to process data iteratively

use SAS functions to perform data manipulation and transformations
use the FORMAT procedure to create user-defined formats.

Software Addressed

This course addresses Base SAS software. This course is appropriate for students who are using SAS 9 software.

Course Contents

Introduction

overview of SAS Foundation
course logistics
creating the course data

Efficient SAS Programming

identifying computer resources related to efficiency

Controlling I/O Processing and Memory

SAS DATA step processing
controlling I/O
reducing the length of numeric variables
compressing SAS data sets
using SAS views

Accessing Observations

access methods
accessing observations by number
creating an index
using an index

DATA Step Arrays

introduction to lookup techniques
one-dimensional arrays
multidimensional arrays
loading a multidimensional array from a SAS data set

DATA Step Hash and Hiter Objects

introduction
hash object methods
loading a hash object from a SAS data set

DATA step hiter object

Combining Data Horizontally

DATA step merges and SQL procedure joins
using an index to combine data
combining summary and detail data
combining data conditionally

User-Defined Functions and Formats

user-defined functions
user-defined formats

Learning More

areas of support from SAS
other courses to consider

Combining Raw Data Files Vertically

combining raw data files vertically

SAS Clinical Program

SAS/GRAPH 1: Essentials

Duration: 3 days

This course teaches you how to produce high-quality presentation graphics using SAS/GRAPH software.

Learn how to

design graphs that describe their data in the most appropriate fashion
customize default graph elements such as axes and legends
apply annotations that supplement the features available with SAS/GRAPH procedures
deliver graphs in multiple formats, such as static image files and client-generated images.

Who should attend: SAS users who must generate graphs using SAS program code in batch or noninteractive execution environments such as SAS Stored Processes, or have the need to produce a large quantity of graphs with minimal user interaction

System Requirements

Prerequisites

Before attending this course, you should have completed the *SAS Programming 1: Essentials* course or have equivalent programming experience.

Course Contents

Course Logistics and Overview

course logistics
graphical reporting overview

Graphical Reporting with SAS/GRAPH

graph types produced by SAS/GRAPH
SAS/GRAPH environment
SAS/GRAPH program structure
SAS/GRAPH and the Output Delivery System

Producing Scatter and Line Plots

creating scatter plots
creating line plots
creating plots containing multiple lines
creating other types of plots with individual data points
creating other types of plots with grouped data (self-study)

Producing Bar Charts

creating vertical and horizontal bar charts
creating grouped and subgrouped bar charts

creating block charts (self-study)
creating bar-line charts
creating area bar charts

Producing Pie, Donut, Star, and Other Charts

creating pie charts
creating donut charts (self-study)
creating star charts (self study)
creating radar charts
creating tile charts
creating key performance indicator charts

Customizing Axes and Legends

customizing plot and chart axes
customizing axes with AXIS statements
customizing legends

Customizing the Appearance of SAS/GRAPH Output

common attributes of graphics elements
customizing common graphics elements
customizing plot appearance
customizing chart appearance
customizing axis appearance

Annotating Graphs

creating basic annotations
creating data-dependent annotations

Generating Graphs for Use in Other Applications

creating image files
creating client-rendered graphs
creating clickable graphs for the Web using ODS

Managing and Replaying SAS/GRAPH Output (Self-Study)

storing and naming graphics output
using the GREPLAY procedure in line mode

SAS Report Writing 1: Using Procedures and ODS

Duration: 3 days

This course teaches you how to create detailed tabular, summary, and listing reports. You also learn how to send report output to the major Output Delivery System (ODS) destinations.

Learn how to

- use PROC REPORT to generate tabular detail reports, detail reports with subtotals and grand totals, and detail reports with computed columns
- generate summary reports based on GROUP, ACROSS, or CLASS variables using PROC REPORT and PROC TABULATE
- insert appropriate summary lines into reports
- send report output to the RTF, PDF, and HTML destinations
- change reports using ODS options and style overrides.

Who should attend: SAS programmers, data managers, report writers, and analysts who must generate reports with the REPORT procedure and the TABULATE procedure and who are familiar with basic SAS procedure syntax and logic

System Requirements

Prerequisites

Before attending this course, you should have completed the *SAS Programming 1: Essentials* course or have equivalent programming experience.

Course Contents

Introduction to the Output Delivery System

- course logistics
- sending a report to an ODS destination

Creating Summary Reports with the TABULATE Procedure

- using basic TABULATE procedure statements
- enhancing the table
- adding percentages
- more about picture formats (self-study)

Advanced TABULATE Procedure Topics

- handling missing values with PROC TABULATE
- using ODS STYLE= options with PROC TABULATE
- working with pages and BY groups
- controlling row structure and data subsets
- using multilabel formats with PROC TABULATE
- working with PROC TABULATE and the listing destination

Creating Detail Reports with the REPORT Procedure

- using basic REPORT procedure statements

adding summary lines
computing an additional column
working with PROC REPORT in the listing destination (self-study)

Creating Summary Reports with the REPORT Procedure

defining and using group variables
customizing break lines
defining and using across variables

Advanced REPORT Procedure Topics

using absolute column names with ACROSS usage
working with missing values and PROC REPORT
working with STYLE= overrides with PROC REPORT
enhancements using the CALL DEFINE statement (self-study)
advanced compute block examples (self-study)

Enhancing Reports with ODS

adding options to ODS destination statements
using additional ODS features (self-study)
using cascading style sheets with ODS (self-study)

Style Attributes and their Values

attributes for use with the STYLE= options in PROC REPORT and PROC TABULATE

Processing Database and Spreadsheet Data with SAS/ACCESS Software

Duration: 1.0 day

This course teaches you how to read Oracle, DB2, or Microsoft Access or Excel tables into your SAS programs using SAS/ACCESS software.

Learn how to

- read data in relational databases and Excel workbooks using the LIBNAME statement and using the SQL Pass-Through Facility
- create efficiency techniques for optimizing data access performance
- join data using the DATA step and the SQL procedure
- import Microsoft Access and Excel data using the Import Wizard, and export data using the Export Wizard.

Who should attend: SAS programmers

Prerequisites

Before attending this course, you should be comfortable programming in SAS and Structured Query Language (SQL). You can gain the SQL knowledge from the *SAS SQL 1: Essentials* course. You can gain knowledge of SAS from the *SAS Programming 1: Essentials* course. A working knowledge of your database is helpful.

Course Contents

Accessing Data in a Relational Database or Microsoft Excel Workbook

- understanding databases
- establishing the requirements to connect to a database
- establishing the requirements to connect to an Excel workbook

Using a SAS/ACCESS LIBNAME Statement

- connecting to a database table using SAS/ACCESS LIBNAME engines
- connecting to an Excel workbook
- explaining and applying Open Database Connectivity (ODBC)
- using an embedded LIBNAME statement

Using the SQL Pass-Through Facility

- passing queries to your database management system (DBMS)
- passing non-queries to your DBMS
- comparing the SQL Pass-Through Facility and the SAS/ACCESS LIBNAME engines

Joining Tables

- combining tables using a DATA step merge
- joining tables using the SQL Pass-Through Facility and the SQL procedure
- joining tables from different databases

Importing and Exporting PC File Data

using the Import Wizard to read a Microsoft Access table and an Excel worksheet into a SAS data set

Using the IMPORT procedure to read a Microsoft Access table and an Excel worksheet into a SAS data set

Using the Export Wizard to write a SAS data set and an Oracle table to an Excel workbook

Using the EXPORT procedure to export a SAS data set to an Excel workbook

Creating and Updating SAS/ACCESS Access and View Descriptors (Self-Study)

creating and updating an access descriptor

creating and updating a view descriptor

converting access and view descriptors to SQL procedure views with the CV2VIEW procedure

SAS Clinical Data Integration: Essentials

Duration: 3.0 days

This course introduces the SAS Clinical Data Integration solution, which makes full use of the SAS Data Integration Studio and adds additional functionality related to clinical data integration. This course introduces the essentials for using SAS Data Integration Studio to register source and target tables and to create and manage jobs using transformations to read and manipulate data. Using the clinical features, this course shows how to import CDISC standards (or other standards) for domain structure and contents into the metadata, build clinical domain target table metadata from those standards, create jobs to load clinical domains, to validate the structure and content of the clinical domains based on the standards, and to generate CDISC standard define.xml files describing the domain tables for clinical submissions. Also covered are methods to monitor the status of clinical data integration projects and to promote new domain structures created within clinical projects to customized standards available across the organization.

Learn how to

- register source data and target tables
- create jobs and explore the functionality of the Job Editor
- work with many of the various transformations
- import clinical standards into metadata
- customize clinical domain tables and columns
- create, load, and verify structure and contents of clinical domain target tables
- monitor clinical projects
- promote new standards
- create define.xml files for CDISC clinical submissions.

Who should attend: Clinical Data Integration administrators and clinical data integration developers

Prerequisites

Before attending this course, you should

- be familiar with clinical studies
- have some knowledge of the process of preparing CDISC clinical study submissions
- be familiar with SAS programming basics. You can gain this experience by completing the *SAS Programming 1: Essentials* course.

Course Contents

Introduction to Clinical Data Integration

- purpose and functions of SAS Clinical Data Integration
- understanding the software components comprising SAS Clinical Data Integration
- overview of SAS Clinical Data Integration components in SAS Data Integration Studio

Technical Overview of SAS Data Integration Studio

Case Study Overview and Defining Metadata for Source Tables and Domains

- Clinical Data Integration case study description
- overview of the domain loading jobs
- defining source table metadata
- defining target domain metadata

Creating Jobs to Load Domains

- introduction
- loading the DM (Demographics) domain
- loading the XP (Pain Diary) domain
- loading the SUPPDM (Supplemental Demographics) domain
- loading the QS (Questionnaire) domain

Domain Compliance Checks

- introduction to CDISC-SDTM compliance checks
- compliance checks for the DM domain
- compliance checks for the QS domain
- compliance checks for an externally supplied domain

Generating CRT-DDS Define.xml Files

- creating a standard CRT-DDS Define.xml document
- creating customized CRT-DDS Define.xml documents

Clinical Data Standards Administration Activities

- importing standard domains, domain columns, and compliance check metadata
- customizing a standard domain
- analyzing domain usage and promotion of custom domains
- creating customized compliance checks
- importing controlled terminology

Clinical Study Administration Activities

- creating and modifying terminology packages
- creating default content
- creating a new clinical study