

## SAS 9.4 Base Programming – Performance Based Beta Exam

### Access and Create Data Structures

#### Create temporary and permanent SAS data sets.

- Use a DATA step to create a SAS data set from an existing SAS data set.

#### Investigate SAS data libraries using base SAS utility procedures.

- Use a LIBNAME statement to assign a library reference name to a SAS library.
- Investigate a library programmatically using the CONTENTS procedure.

#### Access data.

- Access SAS data sets with the SET statement.
- Use PROC IMPORT to access non-SAS data sources.
  - Read delimited and Microsoft Excel (.xlsx) files with PROC IMPORT.
  - Use PROC IMPORT statement options (OUT=, DBMS=, REPLACE)
  - Use the GUESSINGROWS statement
- Use the SAS/ACCESS XLSX engine to read a Microsoft Excel workbook.xlsx file.

#### Combine SAS data sets.

- Concatenate data sets.
- Merge data sets one-to-one.
- Merge data sets one-to-many.

#### Create and manipulate SAS date values.

- Explain how SAS stores date and time values.
- Use SAS informats to read common date and time expressions.
- Use SAS date and time formats to specify how the values are displayed.

#### Control which observations and variables in a SAS data set are processed and output.

- Use the WHERE statement in the DATA step to select observations to be processed.
- Subset variables to be output by using the DROP and KEEP statements.
- Use the DROP= and KEEP= data set options to specify columns to be processed and/or output.

## Manage Data

### Sort observations in a SAS data set.

- Use the SORT Procedure to re-order observations in place or output to a new dataset.

### Conditionally execute SAS statements.

- Use IF-THEN/ELSE statements to process data conditionally.
- Use DO and END statements to execute multiple statements conditionally.

### Use assignment statements in the DATA step.

- Create new variables and assign a value.
- Assign a new value to an existing variable.
- Assign the value of an expression to a variable.
- Assign a constant date value to a variable.

### Modify variable attributes using options and statements in the DATA step.

- Change the names of variables by using the RENAME= data set option.
- Use LABEL and FORMAT statements to modify attributes in a DATA step.
- Define the length of a variable using the LENGTH statement.

### Accumulate sub-totals and totals using DATA step statements.

- Use the BY statement to aggregate by subgroups.
- Use first. and last. processing to identify where groups begin and end.
- Use the RETAIN and SUM statements.

### Use SAS functions to manipulate character data, numeric data, and SAS date values.

- Use SAS functions such as SCAN, SUBSTR, TRIM, UPCASE, and LOWCASE to perform tasks such as the tasks shown below.
  - Replace the contents of a character value.
  - Trim trailing blanks from a character value.
  - Search a character value and extract a portion of the value.
  - Convert a character value to upper or lowercase.
- Use SAS arithmetic, financial, and probability functions to create or modify numeric values by using the INT and ROUND functions.
- Create SAS date values by using the functions MDY, TODAY, DATE, and TIME.
- Extract the month, year, and interval from a SAS date value by using the functions YEAR, QTR, MONTH, and DAY.
- Perform calculations with date and datetime values and time intervals by using the functions INTCK, INTNX, DATDIF and YRDIF.

**Use SAS functions to convert character data to numeric and vice versa.**

- Explain the automatic conversion that SAS uses to convert values between data types.
- Use the INPUT function to explicitly convert character data values to numeric values.

**Process data using DO LOOPS.**

- Explain how iterative DO loops function.
- Use DO loops to eliminate redundant code and to perform repetitive calculations.
- Use conditional DO loops.
- Use nested DO loops.

**Restructure SAS data sets with PROC TRANSPOSE.**

- Select variables to transpose with the VAR statement.
- Rename transposed variables with the ID statement.
- Process data within groups using the BY statement.
- Use PROC TRANSPOSE options (OUT=, PREFIX= and NAME=).

**Use macro variables to simplify program maintenance.**

- Create macro variables with the %LET statement
- Use macro variables within SAS programs.

## Error Handling

**Identify and resolve programming logic errors.**

- Use the PUTLOG Statement in the Data Step to help identify logic errors.
- Use PUTLOG to write the value of a variable, formatted values, or to write values of all variables.
- Use PUTLOG with Conditional logic.
- Use temporary variables N and ERROR to debug a DATA step.

**Recognize and correct syntax errors.**

- Identify the characteristics of SAS statements.
- Define SAS syntax rules including the typical types of syntax errors such as misspelled keywords, unmatched quotation marks, missing semicolons, and invalid options.
- Use the log to help diagnose syntax errors in a given program.

**Examine and resolve data errors.**

- Given a SAS program, use the log to determine the reason for a data error.

## Generate Reports and Output

### Generate list reports using the PRINT procedure.

- Modify the default behavior of PROC PRINT by adding statements and options such as
  - use the VAR statement to select and order variables.
  - calculate totals with a SUM statement.
  - select observations with a WHERE statement.
  - use the ID statement to identify observations.
  - use the BY statement to process groups.

### Generate summary reports and frequency tables using base SAS procedures.

- Produce one-way and two-way frequency tables with the FREQ procedure.
- Enhance frequency tables with options.
- Use PROC FREQ to validate data in a SAS data set.
- Calculate summary statistics and multilevel summaries using the MEANS procedure
- Enhance summary tables with options.
- Identify extreme and missing values with the UNIVARIATE procedure.

### Enhance reports system user-defined formats, titles, footnotes and SAS System reporting options.

- Use the LABEL statement to define descriptive column headings.
- Control the use of column headings with the LABEL and SPLIT=options in Proc Print output.

### Generate reports using ODS statements.

- Identify the Output Delivery System destinations.
- Create HTML, PDF, RTF, and files with ODS statements.
- Use the STYLE=option to specify a style template.
- Create files that can be viewed in Microsoft Excel.

### Export data

- Create a simple raw data file by using the EXPORT procedure as an alternative to the DATA step.
- Export data to Microsoft Excel using the SAS/ACCESS XLSX engine.

---

**Note:** All 24 main objectives will be tested on every exam. The 68 expanded objectives are provided for additional explanation and define the entire domain that could be tested.