SAS Base Programming for SAS 9 Exam

Accessing Data

Use FORMATTED and LIST input to read raw data files.

- Distinguish between character and numeric data, between standard and nonstandard numeric data.
- Use the INPUT statement with formatted input to read character, standard and nonstandard fixed-field data: INPUT variable informat;
- Use the INPUT statement with list input to read character, standard and nonstandard free-format data: INPUT variable <$>;

Use INFILE statement options to control processing when reading raw data files.

- Use the INFILE statement DLM= option to specify field delimiters.
- Use the INFILE statement PAD= option to reading variable-length records.
- Use the INFILE statement DSD= option to read data when missing values occur.
- Use the INFILE statement options, FLOWOVER, MISSOVER, TRUNCOVER and STOPOVER, as appropriate to control reading records.

Use various components of an INPUT statement to process raw data files including column and line pointer controls, and trailing @ controls.

- Use pointer controls.
- Use line controls.
- Use trailing @ and double trailing @.

Combine SAS data sets.

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

Access an Excel workbook.

- Use the SAS/ACCESS EXCEL or PCFILES engine to read an .xlsx file OR use the SAS/ACCESS XLSX engine to read an .xlsx file.

Creating Data Structures

Create temporary and permanent SAS data sets.

- Use a DATA step to create a SAS data set from a text file. Example:

```
Data NewData;
   Infile "path.rawdata";
   Input <pointer-control> variable informat;
Run;
```

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

```
 Data Mylib.NewData;
      Set Mylib.OldData;
      <other SAS statements>
   Run;
```

Create and manipulate SAS date values.

- Explain how SAS stores date and time values.
- Use SAS informats to read common date and time expressions.
- Explain the YearCutOff system option as used to interpret two digit year values.
- Use SAS date and time formats to specify how the values are displayed.

Export data to create standard and comma-delimited raw data files.

- Create a simple text file by using the DATA step with the file and put statements.
- Create a simple raw data file by using the EXPORT procedure as an alternative to the DATA step.
- Use the DSD option to create a comma-delimited file that writes out values separated by a delimiter.

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.

Managing Data

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 and DATASET procedures.

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 SELECT/WHEN statements to process data 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 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.

Process data using one-dimensional SAS arrays.

- Explain the concepts of SAS arrays.
- Use SAS arrays to perform repetitive calculations.
- Define the elements in an array using the ARRAY statement
- Use a DO loop to process an array.

Validate and clean data.

- Use PROC FREQ to list unique values, with the nlevel option to show the number of distinct values, with the order=freq to check for duplicate or missing values.
- Use PROC PRINT with the WHERE statement to display observations with invalid values.
- Use PROC MEAN to validate the range of numeric variables.
- Use PROC UNIVARIATE to display extreme observations and missing values and with the ID statement to display the value of identifying variable(s).

Generating Reports

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 through the use of user-defined formats, titles, footnotes and SAS System reporting.

- 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 LISTING files with ODS statements.
- Use the STYLE=option to specify a style template.
- Create files that can be viewed in Microsoft Excel.

Handling Errors

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.

- Use the SAS system option, ERRORS=n, to specify the maximum number of observations for which SAS issues complete error messages written to the log.
- Given a SAS program, use the log to determine the reason for a data error.

Note: All 27 main objectives will be tested on every exam. The 91 expanded objectives are provided for additional explanation and define the entire domain that could be tested.