SAS® Viya® Fundamentals of Programming Exam

Programming in SAS Viya Concepts (10-15%)

Describe the SAS Viya architecture.

- Compute Server vs. Cloud Analytics Server (CAS)
- Serial vs parallel processing
- In-memory processing
- Open source integration

Explain when to use the CAS server for programming tasks.

- Size of data
- Type of SAS procedure used

Managing Data in CAS (25-30%)

Explain how to access and use CAS Libraries (caslibs).

- Establish CAS sessions with the CAS statement.
- Caslib attributes (Session, local, active, personal)
- Properties of the casuser caslib
- Use the CASLIB statement to assign session-scope caslibs
- Assign a libref to a caslib with the LIBNAME statement and CAS engine
- View the contents of a caslib with PROC CASUTIL

Describe how to load data into In-Memory Tables.

- Load data files into memory
- Client-side vs server-side files
- Loading client-side data (PROC CASUTIL)
    - LOAD DATA= statement
- In-memory table scope (Session vs Global, promoting tables)
- Loading server-side data sources (PROC CASUTIL)
    - LOAD CASDATA= statement
    - ALTERTABLE statement
- Alternate data loading methods (DATA step, PROC SQL, PROC IMPORT)

Describe how to save and drop In-Memory Tables.

- SASHDAT files
- PROC CASUTIL
  - SAVE statement
  - DROPTABLE statement
  - Suppressing errors

Describe CAS column data types.

- Properties of character column variable types
  - CHAR
  - VARCHAR()
  - Determine when to use CHAR vs VARCHAR()
- Properties of numeric column variable types
  - DOUBLE
  - INT32
  - INT64
- Create varchar column variables with the LENGTH statement
- Determine appropriate column data types for example data

DATA Step and SQL programming in CAS (20-25%)

Explain how SAS determines where code executes.

- Location of the input/output data
- What procedures are being run
- What statements/functions are used
- SESSREF= option on the DATA statement
- SESSREF= option within FedSQL
- MSGLEVEL= system option

Explain threading within the SAS DATA step.

- Where code executes: CAS, Compute Server
- Effect of threads on the DATA step
- _THREADID_ and _NTHREADS_ automatic variables
- SINGLE= DATA step option
- Adjust DATA Step code when accumulating totals
- Explain how BY groups are processed in CAS enabled DATA step code
  - Relationship between the distribution of threads and BY GROUP variables
  - DATA step BY GROUP processing and sorting

Update DATA step code to run in CAS.

- DESCENDING keyword
- WHERE= option
- INFILE/INPUT/DATALINES statements
- MODIFY/REMOVE/REPLACE statements
- DATALIMIT= option
- Functions not supported in CAS (Examples: RANBIN, RANUNI, SYMGET, FILEREF, GIT functions)

Update PROC SQL code to run as PROC FEDSQL code.

- Data types
- Supported statements
- Mnemonics vs operators
- SESSREF= option
- Remerge
- Calculated keyword
- SET operations
- Correlated subqueries
- Dictionary tables
- Views
- LIMIT clause
- FORMAT, LABEL vs PROC CASUTIL ALTERTABLE CASDATA statement

CAS-Enabled Procedures and User Defined Formats (20%-25%)

Identify common procedures that run only on the Compute Server.

- PROC FREQ and UNIVARIATE
- SG Graphics procedures
- Use the log file to identify where code executed

Use common procedures that run in both the CAS and Compute Server.

- How SAS determines where the procedure runs
  - Location of the input/output data
  - Which functions/options are used in the code
- PROC MEANS & PROC SUMMARY
  - Common Supported Statements: CLASS/BY/VAR/WHERE/FORMAT
  - Common Supported Statistics: N, NMISS, MIN, MAX, RANGE, MEAN, SUM, STDERR, VAR)
  - Common Unsupported Statistics: MEDIAN, MODE, percentiles
- PROC TRANSPOSE
- BY GROUP processing in CAS
- Use the log file to identify where code executed

Use Common summary procedures that run only in CAS.

- PROC FREQTAB
  - TABLE statement
  - BY statement
- PROC MDSUMMARY
  - VAR statement
  - OUTPUT statement
  - GROUPBY statement

Discuss how user-defined formats are used and stored in CAS.

- Location where formats are stored within CAS
- Saving formats to caslibs with the CASFMTLIB= option
- Save formats to and retrieve from permanent SASHDAT files with a CAS statement
- Assigning formats to in-memory tables

CAS Language (CASL) Programming (10-15%)

Describe the CASL programming language.

- Action Sets
- Actions
- Parameters

Work with tables using CAS action sets.

- TABLE action set
  - Load tables into memory with the loadTable action
  - Explore table attributes with the tableInfo and columnInfo actions
  - Explore table data with the fetch action
- SIMPLE action set
  - Extract unique values with the distinct action
  - Determine category distribution with the freq action
  - Produce descriptive statistics with the summary action
    - Table parameter
    - Input parameter
    - Subset parameter
    - casOut parameter

Note: All 16 main objectives will be tested on every exam. The additional details provide for additional explanation and define the entire domain that could be tested.