SAS Institute

A00-232

SAS 9.4 Advanced Programming -- Performance-Based

QUESTION & ANSWERS

Question: 1

When is a Cartesian product returned?

A. When join conditions are not specified in a PROC SQL set operation.

B. When the keyword ALL is used with the OUTER UNION operator.

C. When join conditions are not specified in a PROC SQL join.

D. When more than two tables are specified in a PROC SQL join.

Answer: C

Explanation/Reference:

A Cartesian product is returned when join conditions are not specified in a PROC SQL join. In a Cartesian product, each row from the first table is combined with every row from the second table.

Question: 2

Which of the following statements is false?

A. The values of macro variables can be up to 65,534 characters long.

B. Macro variables are always user-defined, and their values remain constant until they are changed by the user.

C. Macro variables are text strings that are independent of SAS data sets.

D. A macro variable can be defined and referenced anywhere in a SAS program except within data lines.

Answer: B

Explanation/Reference:

Macro variables are always text strings that are independent of SAS data sets. The value of a macro variable can be up to 65,534 characters long, and the name of a macro variable can be up to 32 characters long. A macro variable can be defined or referenced anywhere in a SAS program except within data lines. There are two types of macro variables: automatic and user-defined.

Question: 3

What does a component object consist of?

A. None of the above.

B. Methods

C. Attributes

D. Both a and b.

Explanation/Reference:

Component objects are data elements that consist of attributes and methods. Attributes are the properties that specify the information while methods define the operations that an object can perform.

Question: 4

The SYMPUTX routine cannot do which of the following things?

A. automatically convert a numeric value to a character value when used to assign a value to a macro variable in a DATA step

B. be used to assign a data set variable as a value to a macro variable

C. be used in a procedure to store a calculated value

D. create a series of macro variables in one DATA step

Explanation/Reference:

The SYMPUTX routine enables you to assign a data set variable as the value of a macro variable. You can also use the SYMPUTX routine to create a series of related macro variables. Because all macro variable values are character strings, SYMPUTX automatically converts any numeric value that you attempt to assign as a value for a macro variable.

Question: 5

Which PICTURE statement displays a nine-digit account number with six leading asterisks followed by the last five digits, such as ******56789?

A. picture actfmt low-high='999999999' (prefix='******');

B. picture actfmt (prefix='******') low-high='99999';

C. picture actfmt (prefix='*****') low-high='999999999';

D. picture actfmt low-high='99999'(prefix='******');

Explanation/Reference:

The five-digit digit selector truncates the leading digits, and the PREFIX= option specifies a character string to place in front of the formatted value. The PREFIX= option must be specified after the template.

Question: 6

Case Study 1: Write your own SAS code in SAS programming environment.
Directions
This scenario uses the Certadv.All data set. Write a SAS program that does the following:
• In a PROC FCMP step, create a temporary custom function called Adding with a numeric argument named Val. The function itself returns the value of Final, which is the value of 38 added to the value of Val. Store the function in the output library Work with the table name of Function and the package name of Add.
• In a global statement, add the appropriate option so that SAS knows where to search for the custom function.
• In a DATA step, create the temporary data set named Work.StudentCost by reading in the data set Certadv.All. Create a custom function for the Fee variable to create a variable named Final_Cost.
• In a PROC PRINT step, create a report based on Work.StudentCost with the following variables displayed:
Student_Name, Course_Code, Fee, and Final_Cost.
Test Your Code
1. What is the value of Final_Cost for observation 144?
2. What is the value of Final_Cost for observation 282?
Exam Objective
Create custom functions with the FCMP procedure.

Answer: NA

Explanation/Reference:

Note: On the live exam, you will be evaluated both on the results of your code and the code itself. Your code should be similar to the following example code, but does not need to match exactly:

```
proc fcmp outlib=work.function.add; /*1*/
   function adding(val); /*2*/
      final=38+val;
      return(final); /*3*/
   endsub; /*4*/
run;
options cmplib=work.function; /*5*/
data work.studentcost; /*6*/
   set certadv.all;
   Final_Cost=adding(fee); /*7*/
run;
proc print data=work.studentcost; /*8*/
   var Student_Name Course_Code Fee Final_Cost;
run;
```

1 The FCMP procedure enables you to create custom functions using DATA step syntax. The OUTLIB= option specifies Work.Function as the table in which the Add package is stored. The Add package is a collection of routines that have unique names.
2 The FUNCTION statement specifies the function name and the function arguments, as well as whether the function returns a character or numeric value. The Adding custom function has one numeric argument named Val, and it returns a numeric value. The variable Final is computed as 38 plus the value of the function's argument.
3 The RETURN statement specifies the value of Final to be returned from the function.
4 The ENDSUB statement ends the syntax for the function.
5 The CMPLIB= option specifies Work.Function table for SAS to search for a package that contains the desired function.

6 The DATA step creates a temporary data set named Work.StudentCost. SAS reads Certadv.All data set to create Work.StudentCost.
7 The DATA step creates a new variable named Final_Cost. The Final_Cost encompasses the fee for the course and an additional student fee. The new variable references the custom function Adding. The function adds a constant value of 38 to the value of Fee. The function returns the value of Final, which is then assigned to the value of Final_Cost.
8 The PROC PRINT step displays the output data with only the specified variables: Student_Name, Course_Code, Fee, and Final_Cost.
Test Your Code Solution
1. Correct Answer: 438
2. Correct Answer: 688

Question: 7

Case Study 2: Write your own SAS code in SAS programming environment.
Directions
This scenario uses the Certadv.Monsal data set. Write a SAS program that does the following:
• Use PROC FORMAT to create a numeric format that specifies a template for displaying month values.
• If the value is 1 through 12, display the value as a two-digit number (month values 1 through 9 should contain a leading zero).
• If the value is not a number between 1 and 12 inclusive, display the value 'Not a valid month'.
• Name the custom format Monthfmt.
• Use a PROC PRINT step to create a report based on Certadv.Monsal. Apply the Monthfmt custom format to the Month variable.
Test Your Code
How many instances of 'Not a valid month' does your output contain?
Exam Objective
Specify a template using the PICTURE statement within the FORMAT procedure.

Answer: NA

Explanation/Reference:

Note: On the live exam, you will be evaluated both on the results of your code and the code itself. Your code should be similar to the following example code, but does not need to match exactly:
```
proc format;
    picture monthfmt
        1-12='99' /*1*/
        other ='Not a valid month'; /*2*/
run;
proc print data=certadv.monsal;
    format month monthfmt.; /*3*/
run;
```
1 The PICTURE statement creates a template called Monthfmt in which it specifies that characters in a range of 1 through 12 are numeric. Using digit selectors (99) specifies the positions for the numeric values. Using a digit selector of 1 through 9 guarantees a leading zero for the month values of 1 through 9. If a digit selector of 0 is used, then there are no leading zeros.
2 The PICTURE statement also adds another definition to the template. It specifies that if the value is not between 1 through 12 inclusive, display the value Not a valid month.
3 The PROC PRINT step displays the data set Certadv.Monsal. You can use the FORMAT statement to apply the custom format Monthfmt to the Month variable.
Test Your Code Solution
Correct Answer: 7