**Quiz, Lesson 1: Getting Started with SAS Programming**

Which of the following lists the steps in the programming process correctly?

1. Define the business need.

2. Write a SAS program.

3. Run the program.

4. Review the results.

5. Debug or modify.

Explain: First, you establish the business need for your program. Then you go through the process of writing and submitting the program, checking your results, and making any necessary changes. You might have to perform this iterative process more than once.

Are raw data files created only by SAS?

b. No

Using SAS, you can read any kind of data.

a. True

What does a SAS program file contain?

a. SAS programming code

Are SAS data sets created only by SAS?

a. Yes

Can a SAS program be saved and reused?

a. Yes

# Quiz, Lesson 2: Working with SAS Programs

Which of the following can represent a step boundary?

a. a RUN statement

b. a QUIT statement

c. a DATA statement

d. a PROC statement

e. all of the above

What does a DATA step typically create?

c. SAS data set

What does a PROC step typically create?

d. report

Which of the following represents a syntax error? Select all that apply.

b. invalid options

c. missing semicolon

f. unmatched quotation marks

h. a misspelled keyword

Which of the following samples of code is valid?

Top of Form

|  |  |  |
| --- | --- | --- |
|  | *b.*   |  | | --- | | title 'New Sales Employees';  proc print data=work.NewSalesEmps;  run; | |

How many step boundaries does this program contain?

data work.staff;

length First\_Name $ 12

Last\_Name $ 18

Job\_Title $ 25;

infile "&path/newemployees.csv" dlm=',';

input First\_Name $ Last\_Name$

Job\_Title $ Salary;

run;

proc print data=work.staff;

run;

proc means data=work.staff;

var Salary;

run;

c. six

Explain: RUN, QUIT, DATA, and PROC statements function as step boundaries, which determine when SAS statements take effect and indicate the end of the current step or the beginning of a new step.

Which of the following is a SAS syntax requirement?

d. End each statement with a semicolon.

Explain: SAS statements usually begin with an identifying keyword, and they must end with a semicolon. Although it is recommended to end steps with a RUN statement, it is optional. The other listed items are related to formatting your programs to make them easier to read.

Which of the following steps is typically used to generate reports and graphs?

b. PROC

Explain: PROC steps are typically used to process SAS data sets (that is, generate reports, graphs, and statistics).

Does this comment contain syntax errors?

/\*

Report created for budget

presentation; revised October 15.

\*/

proc print data=work.newloan;

run;

a. No. The comment is correctly specified.

What result would you expect from submitting this step?

proc print data=work.newsalesemps

run;

b. an error message in the log

Explain: There is a missing semicolon following the data set name. When this step runs, SAS will interpret the word RUN as an option in the PROC PRINT statement (because of the missing semicolon). As a result, the PROC PRINT step will not execute and an error message will be displayed in the log.

SAS allows either single or double quotation marks. If you begin with a single quotation mark, you can end with a double quotation mark.

b. False

What happens if you submit the following program?

porc print data=work.newsalesemps;

run;

b. SAS assumes that the keyword PROC is misspelled and executes the PROC PRINT step.

Where do global statements appear?

a. anywhere in a SAS program

SAS permits your programs to be free format.

a. True

A comment is text in your program that SAS processes along with the other SAS steps in your program.

b. False

Explain: A comment is text in your program that SAS ignores during processing, but writes to the SAS log.

# Quiz, Lesson 3: Accessing Data

Which of the following values can be stored in the variable **Product\_Line**, based on the attributes shown in this partial PROC CONTENTS output? Select all that apply.

| **Alphabetic List of Variables and Attributes** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **#** | **Variable** | **Type** | **Len** | **Format** | **Label** |
| **3** | Product\_Category | Char | 25 |  | Product Category |
| **4** | Product\_Group | Char | 25 |  | Product Group |
| **1** | Product\_ID | Num | 8 | 12. | Product ID |
| **2** | Product\_Line | Char | 20 |  | Product Line |
| **5** | Product\_Name | Char | 45 |  | Product Name |
| **6** | Supplier\_Country | Char | 2 |  | Supplier Country |
| **8** | Supplier\_ID | Num | 8 | 12. | Supplier ID |
| **7** | Supplier\_Name | Char | 30 |  | Supplier Name |

Top of Form

|  |  |
| --- | --- |
|  | *a.*  Dubby Low Men's Street Shoes |
|  | *b.*  Shoes |
|  | *c.*  13198 |
|  | *d.*  A-Team, Kids |
|  | *e.*  Trois Socks (Cush) |

Explain: Bottom of Form

All of the values are valid character values, but the first value is longer than the specified length of 20.

Which statement about SAS libraries is true?

b. A SAS library is a collection of one or more SAS files that are referenced and stored as a unit.

In this SAS program, is **salesbonus** a temporary SAS data set?

|  |
| --- |
| proc means data=salesbonus;  class Job\_Title;  var Amount;  run; |

Top of Form

|  |  |
| --- | --- |
|  | *a.*  yes |

Which of the following librefs is invalid?   
Top of Form

|  |  |
| --- | --- |
|  | *a.*  **\_orionstar** |
|  | *c.*  **or\_01** |
|  | *d.*  **1\_or\_a** |

Which PROC CONTENTS step prints only general information about a SAS library and a listing of the members of the library?

|  |  |  |
| --- | --- | --- |
|  | *b.*   |  | | --- | | proc contents data=orion.\_all\_ nods;  run; | |

Type the correct letter to match the types of information with the portion of a SAS data set in which each is documented or stored.   
Top of Form

|  |  |  |  |
| --- | --- | --- | --- |
|  | the value of **Salary** for observation 1 |  | a. descriptor portion |
|  | the name of the data set |  | b. data portion |
|  | the type of the **Salary** variable |  |  |
|  | the creation date of the data set |  |  |

How many observations and variables does the data set shown here contain?

|  |  |  |
| --- | --- | --- |
| **Company** | **Region** | **Sales** |
| A&MRadio | N | 63500 |
| Jack's TV | S | 45800 |
| Sound City | S | 38900 |
| Music Ltd. | N | 99500 |

Top of Form

|  |  |
| --- | --- |
|  | *b.*  four observations, three variables |

Explain: The SAS data set contains four observations and three variables. Recall that in SAS, observations are the rows in a data set, and variables are the columns in a data set.

In this PROC CONTENTS output, what is the default length of the variable **Street\_ID**?

| **Alphabetic List of Variables and Attributes** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **#** | **Variable** | **Type** | **Len** | **Format** | **Label** |
| **6** | Country | Char | 2 |  | Country |
| **3** | Street\_ID | Num | 8 | 12. | Street ID |
| **5** | Sup\_Street\_Number | Char | 8 |  | Supplier Street Number |
| **4** | Supplier\_Address | Char | 45 |  | Supplier Address |
| **1** | Supplier\_ID | Num | 8 | 12. | Supplier ID |
| **2** | Supplier\_Name | Char | 30 |  | Supplier Name |

Top of Form

|  |  |
| --- | --- |
|  | *a.*  8 bytes |

Explain: The default length of numeric variables is 8 bytes. A numeric variable of the default length can hold 16 or 17 significant digits. 32,767 bytes is the largest possible length for a character variable.

Which of the following variable names are valid? Select all that apply.   
Top of Form

|  |  |
| --- | --- |
|  | *a.*  **data5mon** |
|  | *b.*  **5monthsdata** |
|  | *c.*  **data#5** |
|  | *d.*  **five months data** |
|  | *e.*  **five\_months\_data** |
|  | *f.*  **FiveMonthsData** |

Explain: The correct answers are **a, e,** and **f**. Valid variable names begin with a letter or underscore, and continue with letters, numbers, or underscores.

In which portion of a SAS data set are the following found?

* name of the data set
* type of the variable **Salary**
* creation date of the data set

|  |  |
| --- | --- |
|  | *a.*  descriptor portion |

In this PROC CONTENTS output, what is the default length of the variable **Birth\_Date**?

| **Alphabetic List of Variables and Attributes** | | | | |
| --- | --- | --- | --- | --- |
| **#** | **Variable** | **Type** | **Len** | **Format** |
| **4** | Birth\_Date | Num |  |  |
| **8** | Dependents | Num | 8 |  |
| **2** | Employee\_/Gender | Char | 1 |  |
| **5** | Employee\_Hire\_Date | Num | 8 |  |
| **1** | Employee\_ID | Num | 8 | 12. |
| **6** | Employee\_Term\_Date | Num | 8 |  |
| **7** | Marital\_Status | Char | 1 |  |
| **3** | Salary | Num | 8 |  |

 b.  8 bytes

Explain: **Birth\_Date** is a numeric variable, and the default length of numeric variables is 8 bytes.

Which LIBNAME statement has the correct syntax?

libname reports 'filepath/workshop';

Explain: In a basic LIBNAME statement, you specify the keyword LIBNAME, a valid libref, and then the physical name of the library in quotation marks.

Which of the following data sets will be available if you start a new SAS session?

**orion.sales**

Explain: Any library that you create is a permanent library and is available in subsequent SAS sessions. When a data set is in the temporary **work** library, you can optionally use a one-level name. A one-level name consists of only the data set name, such as **sales** or **newsalesemps**. When you specify a one-level name, SAS assumes that the data set is stored in the **work** library because **work** is the default libref.

In this data set, what type of variable is **Employee\_ID**?

|  |  |  |  |
| --- | --- | --- | --- |
| **Obs** | **Employee\_ID** | **Last** | **Salary** |
| 1 | **.** | Ralston | 29250 |
| 2 | 120101 | Lu | 163040 |
| 3 | 120104 | Billington | 46230 |
| 4 | 120105 | Povey | 27110 |
| 5 | 120106 | Hornsey | **.** |

 b.  numeric

Explain: **Employee\_ID** has a missing value that is displayed as a period. A missing value is displayed as a period for numeric variables and as a blank for character variables. Also, numeric values are right-justified and character values are left-justified by default.

What type of data set is the input data set in this PROC PRINT step?

|  |
| --- |
| proc print data=order\_fact;  run; |

|  |
| --- |
| *a.*  temporary |

Explain: When you specify a one-level data set name, SAS assumes the library is **work** and the data set is temporary.

A numeric variable can store numeric values with a maximum of eight digits.

|  |
| --- |
| *b.*  False |

Explain: Numeric values are stored in floating point notation in 8 bytes of storage, allowing a maximum of 16-17 digits.

To disassociate a libref that you previously assigned, you can use the UNASSIGN option in the LIBNAME statement.

|  |
| --- |
| *b.*  False |

Explain: You use the CLEAR option in the LIBNAME statement to cancel, or disassociate, a libref that you previously assigned.

In this data set, what do the values *72*, *64*, *68*, and *76* represent?

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | | **Height** | **Weight** | **Age** |
| 54209 | | 72 | 165 | 35 |
| 38204 | | 64 | 122 | 46 |
| 10462 | | 68 | 154 | 28 |
| 38754 | | 76 | 188 | 51 |
| *c.*  values of a variable | | | | |

Explain: These are the values for the variable **Height**. Recall that in SAS, the observations are the rows in a data set, and variables are the columns in a data set. The values 72, 64, 68, and 76 are displayed in the column for **Height**.

Which of the following librefs is valid?

|  |
| --- |
| *c.*  **Car\_2012** |

Explain: This libref follows all three rules for valid librefs. A libref must have a length of one to eight characters, and must begin with a letter or underscore. The remaining characters must be letters, numbers, or underscores.

# Quiz, Lesson 4: Producing Detail Reports

Which SUM statement will produce column totals for the variables **Quantity** and **Total\_Retail\_Price**?

|  |
| --- |
| proc print data=orion.order\_fact;  var Customer\_ID Order\_Date Quantity  Total\_Retail\_Price;  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;  run; |

Top of Form

|  |  |
| --- | --- |
|  | *c.*  sum Quantity Total\_Retail\_Price; |

Copy and paste the following code, which contains two WHERE statements, into the editor. Submit the code. **Reminder**: Make sure you've [defined the **orion** library](javascript:%20openOther('setup_reminder.htm');).

|  |
| --- |
| proc print data=orion.sales;  where Country='AU';  where Salary<30000;  run; |

Which of the following is true?   
Top of Form

|  |  |
| --- | --- |
|  | *d.*  The program executes, but only the second WHERE condition is applied. |

Explain: The following log message indicates that SAS replaces the first WHERE condition with the second WHERE condition: 

NOTE: WHERE clause has been replaced.

Which WHERE statement correctly subsets on the numeric values for May, June, or July and missing character names?

Top of Form

|  |  |
| --- | --- |
|  | *b.*  where Months in (5,6,7) and Names=' '; |

The values for the variable **Name** in the table below are in the form *last name, first name*. Which WHERE statement will return all the observations that have a first name starting with the letter M for the given values?

|  |  |  |
| --- | --- | --- |
| **Name** | | |
| Elvish, Irenie | | |
| Ngan, Christina | | |
| Hotstone, Kimiko | | |
| Daymond, Lucian | | |
| Hofmeister, Fong | | |
| Denny, Satyakam | | |
| Clarkson, Sharryn | | |
| Kletschkus, Monica | | |
|  | *b.*  where Name like '%, M%'; | |

Explain: By using a percent sign, the pattern specifies last names that contain any number of characters. The last name must be followed by a comma, a space, and an uppercase M to start the first name. This can be followed by any number of characters. If you use an underscore, exactly one character must occupy that position.

Which step sorts the observations in a SAS data set and overwrites the same data set?

Top of Form

|  |  |  |
| --- | --- | --- |
|  | *c.*   |  | | --- | | proc sort data=work.empsau;  by First;  run; | |

Explain: PROC SORT replaces the original data set unless you specify an output data set in the OUT= option.

Which BY statement in a PROC SORT step can produce the output shown here?

| **Obs** | **Postal\_Code** | **Employee\_ID** |
| --- | --- | --- |
| **1** | 92173 | 120807 |
| **2** | 92131 | 120661 |
| **3** | 92129 | 121074 |
| **4** | 92128 | 121128 |
| **5** | 92128 | 120755 |
| **6** | 92128 | 120730 |
| **7** | 92126 | 121049 |
| **8** | 92124 | 121029 |
| **9** | 92124 | 121021 |
| **10** | 92122 | 120744 |

Top of Form

|  |  |
| --- | --- |
|  | *d.*  by descending Postal\_Code descending Employee\_ID; |

Explain: In the output, the observations are sorted in descending order for **Postal\_Code** and, within each postal code, in descending order for **Employee\_ID**. The BY statement must specify the keyword DESCENDING before each variable.

Which footnote or footnotes appear in the second procedure results?

|  |
| --- |
| footnote1 'Orion Star';  footnote2 'Sales Employees';  footnote3 'Confidential';  proc print data=orion.sales;  run;  footnote2 'Non Sales Employees';  proc print data=orion.nonsales;  run; |

Top of Form

|  |  |
| --- | --- |
|  | *b.*  Orion Star        Non Sales Employees |

Explain: When you run the second PROC PRINT step, the FOOTNOTE2 statement replaces the previous footnote with the same number: Non Sales Employees replaces Sales Employees. It also cancels all footnotes with higher numbers, so FOOTNOTE3, Confidential, does not appear in the results. The resulting footnotes are Orion Star and Non Sales Employees.Bottom of Form

Which observation or observations will be selected by the following WHERE statement?

|  |  |  |
| --- | --- | --- |
| where Job\_Title contains 'I'; | | |
| **Obs** | **Last\_Name** | **First\_Name** | | **Country** | **Job\_Title** |
| 1 | Wu | Christine | | AU | Sales Rep I |
| 2 | Stone | Kimiko | | AU | Sales Manager |
| 3 | Hoffman | Fred | | AU | Insurance Sales |

|  |
| --- |
| *d.*  observations 1 and 3 |

Explain: Expressions in the WHERE statement are case sensitive. This WHERE statement returns only those values that contain the exact character string shown. The position of the substring within the value is not important.

You want the variables **Birth\_Date** and **Employee\_Hire\_Date** to be labeled in your PROC PRINT report. What do you need to do to this program?

|  |
| --- |
| proc print data=orion.employee\_payroll;  var Employee\_ID Birth\_Date Employee\_Hire\_Date Salary;  label Birth\_Date='Birth Date'  Employee\_Hire\_Date='Hire Date';  run; |

|  |
| --- |
| *d.*  Add the LABEL option to the PROC PRINT statement. |

Explain: The LABEL statement is not a global statement. You can specify labels for multiple variables in one LABEL statement, or you can use a separate LABEL statement for each variable. To display temporary labels in your PROC PRINT report, you use the LABEL option in your PROC PRINT statement because PROC PRINT doesn't display labels automatically.

Which statement about this PROC SORT step is true?

|  |
| --- |
| proc sort data=orion.staff;  out=work.staff\_sort;  by descending Salary  Manager\_ID;  run; |

|  |
| --- |
| *c.*  A semicolon should not appear after the input data set name. |

Explain: This PROC SORT step has a syntax error: a semicolon in the middle of the PROC SORT statement. If you correct this syntax error, this step sorts**orion.staff** by **Salary** in descending order and by **Manager\_ID** in ascending order. The step then creates the temporary data set **work.staff\_sort**that contains the sorted observations and all variables.

Which of the following statements selects from a data set only those observations for which the value of the variable **Style** is *RANCH*, *SPLIT*, or*TWOSTORY*?

|  |
| --- |
| *d.*  where Style in ('RANCH','SPLIT','TWOSTORY'); |

Explain: In the WHERE statement, the IN operator enables you to select observations based on several values. You specify values in parentheses and separate them by spaces or commas. Character values must be enclosed in quotation marks and must be in the same case as in the data set.

Which of the following statements selects rows in which **Amount** is less than or equal to *$5,000* or **Rate** equals *0.095*?

|  |
| --- |
| *a.*  where amount <= 5000 or rate=0.095; |
| *b.*  where amount le 5000 or rate=0.095; |
| *c.*  where amount <= 5000 or rate eq 0.095; |
| *d.*  all of the above |

Explain: All of the statements shown here select rows in which **Amount** is less than or equal to $5000 or **Rate** equals 0.095.

When you run this code, which title or titles appear in the last PROC PRINT output?

|  |
| --- |
| title1 'The First Line';  title2 'The Second Line';  proc print data=orion.sales;  run;  title2 'The Next Line';  proc print data=orion.sales;  run;  title 'The Top Line';  proc print data=orion.sales;  run; |

|  |
| --- |
| *a.*  The Top Line |

Explain: The TITLE statement in the last PROC PRINT step changes the first title line and cancels all previously specified titles with line numbers higher than 1.

Which program creates the output shown here?

| **Obs** | **EMP ID** | **Employee Hire Date** |
| --- | --- | --- |
| 1 | 120101 | 01JUL2001 |
| 2 | 120102 | 01JUN1989 |
| 3 | 120103 | 01JAN1974 |
| 4 | 120104 | 01JAN1981 |
| 5 | 120105 | 01MAY1999 |
| 6 | 120106 | 01JAN1974 |

|  |
| --- |
|  |
| *b.*   |  | | --- | | proc print data=orion.staff split='+';  var Employee\_ID Emp\_Hire\_Date;  label Employee\_ID='Emp ID'  Emp\_Hire\_Date='Employee+Hire Date';  run; | |

Explain: To display temporary labels in PROC PRINT output, you must specify either the SPLIT= option or the LABEL option in the PROC PRINT statement.

Which BY statement is valid for this PROC PRINT step?

|  |
| --- |
| proc sort data=orion.staff  out=work.staff\_sort;  by Gender Start\_Date;  run;  proc print data=work.staff\_sort label;  by \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;  label Start\_Date='Start';  run; |

|  |
| --- |
| *d.*  by Gender; |

Explain: You can group by **Gender** because **Gender** is the first variable that the data set is sorted by. You could also group by **Gender** and then**Start\_Date** because the data set is sorted by these variables in the same order. You cannot use the other BY statements because the variables are not specified for grouping in the same order that the data set is sorted.

Suppose you already ran the first program, which created a one-page report. Next, you want to run the second program. What will appear at the top of the second report?

|  |
| --- |
| title1 'RADIX Company';  title3 'DVD Sales';  proc print data=radix.sales;  where UnitSold>=30;  run; |

|  |
| --- |
| title2 'Best Sales;  title;  proc print data=radix.staff;  where Sales>25000;  run; |

|  |
| --- |
| *a.*  no titles |

Explain: No titles will appear at the top of the second PROC PRINT report. The null TITLE statement above that statement cancels all previously specified titles.

Which statement about this program is true?

|  |
| --- |
| proc print data=orion.sales;  var Employee\_ID Salary;  where Country='AU';  by Gender;  label Salary='Annual Salary';  run; |

|  |
| --- |
| *b.*  The PROC PRINT report displays only the observations in which the value of **Country** is *AU*. |

Explain: The WHERE statement subsets the data so that the report displays only the observations in which the value of **Country** is AU. The input data set must be sorted by **Gender**, the variable specified in the BY statement. The LABEL and FORMAT statements affect only the report output; they do not affect the input data set.Bottom of Form

Bottom of Form

# Quiz, Lesson 5: Formatting Data Values

Which format creates the displayed value shown here?  
$5,950.35

Top of Form

|  |  |
| --- | --- |
|  | *c.*  DOLLAR9.2 |

Explain: The DOLLARw.d format writes numeric values with a leading dollar sign, a comma that separates every three digits, and a period that separates the decimal fraction. The displayed value is nine characters wide, so the total format width, w, is set to 9. This includes the special characters and decimal places. The displayed value contains two decimal places, so dis set to 2.

Which FORMAT statement formats the variable values as shown below?

|  |  |  |
| --- | --- | --- |
| **Birth\_Date** | **Emp\_Hire\_Date** | **Emp\_Term\_Date** |
| 28/09/1968 | 01/10/1989 | 01/31/09 |

Top of Form

|  |  |
| --- | --- |
|  | *c.*  format Birth\_Date Emp\_Hire\_Date ddmmyy10. Emp\_Term\_Date mmddyy8.; |

Explain: The variables **Birth\_Date** and **Emp\_Hire\_Date** are both displayed as a two-digit day, a two-digit month, and a four-digit year; the day precedes the month. The displayed values have a length of 10. This is the DDMMYY10. format.   
The variable **Emp\_Term\_Date** is displayed as a two-digit month, a two-digit day, and a two-digit year; the month precedes the day. The displayed value has a length of 8. This is the MMDDYY8. format.   
There is no DDMMYYYY or MMMYYYY SAS format.

Which user-defined format names are valid? Select all that apply.   
Top of Form

|  |  |
| --- | --- |
|  | *a.*  $STFMT |
|  | *c.*  \_4YEARS |
|  | *d.*  SALRANGES |

Explain: Character formats begin with a dollar sign and must be followed by a letter or underscore. Answer choice b has a dollar sign followed by a number. Also, user-defined formats cannot be the name of a SAS format, as in answer choice e.

How will a value of *50000* be displayed if the TIERS format below is applied to the value?

|  |
| --- |
| proc format;  value tiers 20000-<50000 ='Tier1'  50000-<100000='Tier2'  100000-250000='Tier3';  run; |

Top of Form

|  |  |
| --- | --- |
|  | *b.*  Tier2 |

Explain: In Tier1, the less-than symbol is before 50000, which means that value will be excluded from the range. In Tier2, however, 50000 is the starting value of the range and will be included in the tier.

Can you include multiple VALUE statements in a single PROC FORMAT step?

|  |
| --- |
| proc format;  value $ctryfmt 'AU'='Australia'  'US'='United States'  other='Miscoded';  value tiers low-<50000 ='Tier1'  50000-<100000='Tier2'  100000-high ='Tier3';  run; |

Top of Form

|  |  |
| --- | --- |
|  | *a.*  yes |

Explain: You can create multiple user-defined formats in the same PROC FORMAT step by specifying multiple VALUE statements.

Which of the following is a valid name for a character format?

|  |
| --- |
| *b.*  $ctry |

Explain: Character formats must start with a dollar sign followed by a letter or underscore. A format name does not end with a period. The period is a required delimiter when using a format in a FORMAT statement.

You specify the variable to which a format applies when you create it in a PROC FORMAT step.

|  |
| --- |
| *b.*  False |

Explain: Formats are not associated with a specific variable until they are applied with a FORMAT statement.

Which of the following FORMAT statements was used to create this output?

| **Employee\_ID** | **Job\_Title** | **Salary** | |
| --- | --- | --- | --- |
| 120102 | Sales Manager | $108255.00 | |
| 120103 | Sales Manager | $87,975.00 | |
| 120121 | Sales Rep. II | $26,600.00 | |
| 120122 | Sales Rep. II | $27,475.00 | |
| 120123 | Sales Rep. I | $26,190.00 | |
| *d.*  format Salary dollar10.2; | | |

Explain: The data value in the first observation was displayed without commas because the format width was not large enough. It contains 10 characters, indicating a format width of 12 with two decimal places.

Which of the following FORMAT statements was used to create this output?

| **Obs** | **Order\_ID** | **Order\_Date** | **Delivery\_Date** |
| --- | --- | --- | --- |
| 1 | 1230058123 | 11JAN07 | 01/11/07 |
| 2 | 1230080101 | 15JAN07 | 01/19/07 |
| 3 | 1230106883 | 20JAN07 | 01/22/07 |
| 4 | 1230147441 | 28JAN07 | 01/28/07 |
| 5 | 1230315085 | 27FEB07 | 02/27/07 |

|  |
| --- |
| *b.*  format Order\_Date date7. Delivery\_Date mmddyy8.; |

Explain: The DATE7. format displays a two-digit day, three-letter month abbreviation, and two-digit year. The MMDDYY8. format displays a two-digit month, day, and year, separated by slashes.

Which of the following is not true of SAS date values?

|  |
| --- |
| *d.*  The base date is January 1, 1900. |

Explain: All of these are true of SAS date values except that SAS stores date values as the number of days between January 1, 1960, and a specific date.

You can use either < or > to define a non-inclusive range in a VALUE statement.

|  |
| --- |
| *b.*  False |

Explain: You can only use the < symbol to define a non-inclusive range.

The format name must include a period delimiter in the FORMAT statement.

|  |
| --- |
| *a.*  True |

Explain: The period is a required syntax element in a format name within a FORMAT statement.

You want to apply the following user-defined format to the numeric variable **Age**. The values of **Age** are stored with one decimal place. Which of the following statements is true regarding the PROC FORMAT step?

|  |
| --- |
| proc format;  value $agegp low-65='Non Retirement'  66<-high='Retirement';  run; |

|  |
| --- |
| *c.*  The format name does not match the variable type. |

Explain: A user-defined format that applies to numeric values cannot start with $. The first value range is an inclusive range, which means it includes the first value and the last value, so 65 will be included in the range. The value 66 will not be displayed as Retirement because the less-than symbol appears directly after it, so it will be excluded from the range. The formatted value is always a character string, no matter whether the format applies to character values or numeric values. A character string can consist of any type of character.

A format modifies both the stored value and the displayed value.

|  |
| --- |
| *b.*  False |

Explain: A format affects the way a value is displayed. It does not change the stored value in any way.

A FORMAT statement is used only to apply SAS formats.

|  |
| --- |
| *b.*  False |

Explain: A FORMAT statement is used to apply both SAS and user-defined formats.

# Quiz, Lesson 6: Reading SAS Data Sets

Type the letter of the word or phrase on the right that completes the statements on the left.   
Top of Form

|  |  |  |  |
| --- | --- | --- | --- |
|  | When you submit a DATA step, SAS processes the step in the \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ phase first. |  | a. descriptor portion |
|  | When you submit a DATA step, SAS processes the step in the \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ phase second. |  | b. compilation |
|  | During the compilation phase, SAS creates the \_\_\_\_\_\_\_\_\_\_\_\_\_ of the output data set. |  | c. execution |
|  | During the execution phase, SAS creates the \_\_\_\_\_\_\_\_\_\_\_\_\_ of the output data set. |  | d. data portion |

Explain: The compilation phase precedes the execution phase. SAS creates the descriptor portion of the data set during the compilation phase and the data portion during the execution phase.

Copy and paste this program, which includes a WHERE statement to subset on the **Bonus** amount, into the editor and submit it. **Reminder**: Make sure you've [defined the **orion** library](javascript:%20openOther('setup_reminder.htm');).

|  |
| --- |
| data work.subset1;  set orion.sales;  Bonus=Salary\*.10;  where Country='AU' and  Bonus>=3000;  run;  proc print data=work.subset1;  run; |

Is the output data set created successfully?   
Top of Form

|  |  |
| --- | --- |
|  | *b.*  no |

Explain: No, the output data set is not created successfully. The log contains an error message, and SAS stopped processing the step. Because **Bonus** is a new variable being created in this DATA step and is not in **orion.sales**, it cannot be used in a WHERE statement. SAS stopped processing the DATA step.

When you use the subsetting IF statement, how are observations excluded?   
Top of Form

|  |  |
| --- | --- |
|  | *b.*  If the expression is false, SAS excludes the observation from the output data set. |

Explain: When the expression is false, SAS excludes the observation from the output data set and continues processing.

Select the situation(s) in which you can use the WHERE statement to subset observations. Select all that apply.   
Top of Form

|  |  |
| --- | --- |
|  | *a.*  in a PROC step |
|  | *b.*  in a DATA step, when the variable in the condition is created |
|  | *c.*  in a DATA step, when the variable in the condition is in the input data set |

Explain: You can use a WHERE statement to subset observations in situations **a** and **c**. A subsetting IF statement can be used in situations **b** and **c**.

If you submit this program, which of the following column headings will display for **Job\_Title** in the resulting report?

|  |
| --- |
| data work.us;  set orion.sales;  where Country='US';  Bonus=Salary\*.10;  label Job\_Title='Sales Title';  drop Employee\_ID Gender Country Birth\_Date;  run;  proc print data=work.us label;  label Job\_Title='Title';  run; |

Top of Form

|  |  |
| --- | --- |
|  | *c.*  Title |

Explain: The column heading will be **Title**, the label specified in the PROC PRINT step. Labels and formats that you specify in PROC steps override the permanent labels in the current step. However, the permanent labels are not changed.Bottom of Form

What statement is used to read a SAS data set in a DATA step?

|  |
| --- |
| *c.*  SET statement |

Explain: A SET statement reads observations from a SAS data set for further processing in the DATA step.Bottom of Form

Bottom of Form

Bottom of Form

Bottom of Form

Bottom of Form

Bottom of Form

Bottom of Form

Bottom of Form

Bottom of Form

Bottom of Form

Bottom of Form

Bottom of Form

Bottom of Form

Bottom of Form

Bottom of Form

Bottom of Form

Bottom of Form

In this PROC PRINT step, which statement can you use to subset observations?

|  |
| --- |
| proc print data=work.us;  run; |

|  |
| --- |
| *a.*  only a WHERE statement |

Explain: To subset observations in a PROC step, you must use a WHERE statement. You cannot use a subsetting IF statement.

What is the name of the output data set in the program below?

|  |
| --- |
| data work.us;  set orion.sales;  where Country='US';  run; |

|  |
| --- |
| *a.* **work.us** |

Explain: The DATA statement provides the name of the SAS data set being created, **work.us**.

Which of the following DATA steps correctly reads the permanent data set **salesinfo** from the **sporting** library and creates a new data set named**salesinfo2** in the same library?

|  |  |
| --- | --- |
| *c.*   |  | | --- | | data sporting.salesinfo2;  set sporting.salesinfo;  run; | |

Explain: The SET statement identifies the input data set, **salesinfo**, which is stored in the permanent library, **sporting**. The DATA statement identifies the output data set, **salesinfo2**, to be created in the permanent library, **sporting**.

Which of the following is **not** created during the compilation phase?

|  |
| --- |
| *b.*  the first observation |

Explain: During compilation, SAS creates the PDV and the descriptor portion of the new data set. SAS creates the first observation during the execution phase.

The data set **orion.sales** contains nine variables. Given this DATA step, how many variables does the descriptor portion of **work.comp** contain?

|  |
| --- |
| data work.comp;  set orion.sales;  drop Gender Salary Birth\_Date;  run; |

|  |
| --- |
| *a.*  six |

Explain: At compile time, SAS uses the descriptor portion of the input data set, **orion.sales**, to create nine variables in the PDV. The DROP statement sets drop flags for three of the nine variables. SAS writes the six variables without drop flags to the output data set, **work.comp**.

The data set **orion.sales** contains nine variables. Given this DATA step, how many variables does the descriptor portion of **work.comp** contain?

|  |
| --- |
| data work.comp;  set orion.sales;  keep Employee\_ID Gender Job\_Title Salary;  run; |

|  |
| --- |
| *a.*  four |

Explain: When you use LABEL and FORMAT statements in a DATA step, SAS permanently associates the labels and formats to the variables. You can override these permanent labels and formats by specifying different labels and formats in a PROC PRINT step. 

|  |
| --- |
| data work.us;  set orion.sales;  label Job\_Title='Sales Title' Hire\_Date='Date Hired';  format Salary commax8. Bonus commax8.2  Hire\_Date ddmmyy10.;  run;  proc print data=work.us label;  label Job\_Title='Title';  format Hire\_Date date9.;  run; |

|  |
| --- |
| *a.*  True |

Explain: Labels and formats that you specify in PROC steps override the permanent labels and formats in the current step. However, the permanent labels and formats are not changed.

What value will be assigned to **Units**?

|  |
| --- |
| data work.comp;  set work.sales;  Units=Total+Bonus/Quantity;  run; |

**Partial PDV**

|  |  |  |  |
| --- | --- | --- | --- |
| **Total** | **Quantity** | **Bonus** | **Units** |
| 140 | 10 | 50 | . |

|  |
| --- |
| *b.*  *145* |

Explain: SAS executes the expression on the right side of the assignment statement following normal operator precedence, and the result is assigned to**Units**. The division occurs first (50/10 is 5), and then the addition occurs (140+5 is 145).

Which procedure can be used to view the permanent labels and formats stored in a data set?

|  |
| --- |
| *a.*  PROC CONTENTS |

Explain: PROC CONTENTS displays the descriptor portion of a data set, and SAS stores permanent labels and formats in the descriptor portion.