# SAS Institute

# A00-215

## SAS Certified Associate: Programming Fundamentals Using SAS 9.4

**For More Information – Visit link below:**

**https://www.examsempire.com/**

**Product Version**

1. **Up to Date products, reliable and verified.**
2. **Questions and Answers in PDF Format.**

# Latest Version: 6.0

## Question: 1

Consider the following SAS code:

```
data work.sales;
  input cust_id $  sales_amt;
  cards;
100 1200
200 850
300 1500
400 900
;
run;

data work.sales_summary;
  set work.sales;
  if sales_amt > 1000 then output;
run;
```

Which of the following statements is TRUE about the SAS code?

A. The code will create two datasets, 'work-sales' and but 'work.sales_summary' will be empty because the statement condition is not met for any records in the 'work.sales' dataset.
B. The code will create two datasets, 'work-sales' and -work.sales_summary' and will contain all records from 'work.sales'
C. The code will create two datasets, 'work-sales' and "work.sales_summary' and 'work.sales_summary' will contain only records where 'sales_amt' is greater than 1000.
D. The code will create a single dataset, 'work-sales' , and 'work.sales_summary' will not be created.
E. The code will produce an error because the 'output' statement is not allowed in a 'data' step.

**Answer: C**

Explanation:
The code will create two datasets: 'work-sales' and 'work-sales_summary' - The 'output' statement in the 'data' step for 'work-sales_summary' is used to control which records are written to the output dataset. The 'if sales_amt > 1000' condition ensures only records where the 'sales_amt' is greater than 1000 are written to 'worksales_summary'.

## Question: 2

You have a SAS dataset ‘work.sales' with variables 'sales_amt', and ‘product id'. You want to create a new dataset that summarizes total sales by 'product id'. Which of the following PROC steps would achieve this?

A.

```
proc summary data=work.sales;
  class product_id;
  var sales_amt;
  output out=work.sales_by_product sum(sales_amt)=total_sales;
run;
```

B.

```
proc means data=work.sales;
  class product_id;
  var sales_amt;
  output out=work.sales_by_product sum(sales_amt)=total_sales;
run;
```

C.

```
proc sql;
  create table work.sales_by_product as
  select product_id, sum(sales_amt) as total_sales
  from work.sales
  group by product_id;
quit;
```

D.

```
proc tabulate data=work.sales;
  class product_id;
  var sales_amt;
  table product_id, sum(sales_amt)=total_sales;
  output out=work.sales_by_product;
run;
```

E.

```
proc freq data=work.sales;
  tables product_id;
  output out=work.sales_by_product sum(sales_amt)=total_sales;
run;
```

**Answer: A,C**

Explanation:

Both PROC SUMMARY and PROC SQL can achieve the desired result PROC SUMMARY uses the ‘class' statement to group data by 'product_id' , and the 'output' statement to create a new dataset with the summary statistics- PROC SQL uses a 'SELECT statement with and 'GROUP BY to achieve the same result Option D, PROC TABULATE, is designed for creating tables, and Option E, PROC FREQ is for frequency tables. Option B,PROC MEANS, can calculate summary statistics but does not have the ability to create a new dataset with the output.

## Question: 3

You are working with a dataset named 'work.customeö that contains a variable called 'age’ representing the age of customers. You want to create a new variable 'age_group' based on the following criteria: 'age_group' = Young' if 'age' is less than 30 'age_group' = 'Middle' if ‘age’ is between 30 and 50 (inclusive) 'age_group' = 'Senior' if 'age' is greater than 50 Which SAS code snippet correctly creates the 'age_group' variable?

A.

```
data work.customer;
  set work.customer;
  if age < 30 then age_group = 'Young';
  if age >= 30 and age <= 50 then age_group = 'Middle';
  if age > 50 then age_group = 'Senior';
run;
```

B.

```
data work.customer;
  set work.customer;
  if age < 30 then age_group = 'Young';
  else if age >= 30 and age <= 50 then age_group = 'Middle';
  else age_group = 'Senior';
run;
```

C.

```
data work.customer;
  set work.customer;
  if age < 30 then age_group = 'Young';
  if age >= 30 and age <= 50 then age_group = 'Middle';
  else age_group = 'Senior';
run;
```

D.

```
data work.customer;
  set work.customer;
  if age < 30 then age_group = 'Young';
  else if age >= 30 then age_group = 'Middle';
  else age_group = 'Senior';
run;
```

E.

```
data work.customer;
  set work.customer;
  if age < 30 then age_group = 'Young';
  else if age > 50 then age_group = 'Senior';
  else age_group = 'Middle';
run;
```

**Answer: A,C**

Explanation:
Both Option A and Option C will create the desired 'age_group' variable using correct logic for the age range. The code in Option A uses separate 'if statements for each condition, while Option C uses 'if statements for the first two conditions and 'else' for the third condition. Both approaches are valid and will produce the same result. Option B and Option D have incorrect logic in their conditional statements. Option E does not correctly assign 'Middle' status to those aged 30-50_ Remember that the 'else' statement is only executed if none of the preceding 'if conditions are true.

## Question: 4

Consider the following SAS code:

```
options nodate nonumber;
data work.new_data;
  input name $ age;
  cards;
John 30
Jane 25
Mary 40
;
run;

proc print data=work.new_data;
run;
```

What is the purpose of the statement options nodate nonumber;?

A. It suppresses the printing of the date and time in the log.
B. It prevents the creation of the output dataset.
C. It alters the default data type for the variables.
D. It specifies the output format for the printed data.
E. It changes the default input delimiter.

**Answer: A**

Explanation:
The statement options nodate nonumber; is a global statement that modifies the SAS environment options nodate suppresses the printing of the date in the log, and options nonumber suppresses the printing of the line number. The other options are incorrect. Option B is incorrect because the output dataset is still created. Option C is incorrect because the default data type for the variables is not changed_ Option D is incorrect because the output format for the printed data is not specified. Option E is incorrect because the default input delimiter is not changed.

## Question: 5

You have a SAS dataset named 'sales' with variables 'product', 'region', and 'units_sold'. You want to create a new dataset 'sales _ summary' that summarizes the total units sold for each product across all regions. Which of the following code snippets correctly achieves this?

A.

```
proc sql;
create table sales_summary as
select product, sum(units_sold) as total_units_sold
from sales
group by product;
quit;
```

B.

```
proc summary data=sales;
class product;
var units_sold;
output out=sales_summary sum=total_units_sold;
run;
```

C.

```
proc means data=sales;
class product;
var units_sold;
output out=sales_summary sum=total_units_sold;
run;
```

D.

```
proc tabulate data=sales;
class product;
var units_sold;
table product, sum(units_sold);
run;
```

E.

```
proc transpose data=sales out=sales_summary;
by product;
var units_sold;
run;
```

**Answer: A,B,C**

Explanation:
All three code snippets (A, B, and C) correctly achieve the desired result of summarizing the total units sold for each product. Option A uses the 'PROC SQL' procedure, which is very flexible for data manipulation and aggregation. It uses the function to calculate the total units sold per product and the 'GROUP BY' clause to group the results by product. Option B uses the 'PROC SUMMARY procedure, which is designed specifically for summarizing data. The 'CLASS' statement specifies the variable to group by, the 'VAR' statement specifies the variable to summarize, and the 'OUTPUT statement defines the output dataset and the calculation to be performed (sum in this case). Option C uses the 'PROC MEANS procedure, similar to 'PROC SUMMARY but focused on generating descriptive statistics. It achieves the desired outcome by specifying the 'CLASS' and 'VAR statements and using the 'OUTPUT' statement to create the output dataset with the sum of units sold per product. Option D uses ' PROC TABULATE which is primarily for creating tables and does not directly produce a dataset with the desired summary Option E uses 'PROC TRANSPOSE to transpose the dataset which is not appropriate for this scenario.

## Question: 6

Which of the following SAS statements would result in an error?

```
1. data work.new_data;
2. input name $ age;
3. cards;
4. John 30;
5. Jane 25;
6. Mary 40;
7. run;

proc print data=work.new_data;
run;
```

Assuming that the above code is in a SAS program, and the program is run from a SAS session.

A. Statement 1 - data worknew data,
B. Statement 2 - input name S age;
C. Statement 4 - John 30,
D. Statement 7 - run;
E. proc print run;

**Answer: C**

Explanation:
Statement 4 - 'John 30;' would result in an error because it is a data line within the CARDS statement and needs to be terminated with a semicolon. In SAS, data lines within the CARDS statement should end with a semicolon (;) to indicate the end of a data line. Without a semicolon, the SAS system interprets the line as incomplete, leading to an error. The other statements are syntactically correct and would execute without errors.

## Question: 7

You are running a SAS program that reads data from a file and performs calculations. The following error message appears in the SAS log:

```
ERROR 22-322: Syntax error, expecting a semicolon.
```

Which of the following is the most likely cause of this error, and where would you look in your code to find the
issue?

A. A missing semicolon at the end of a statement within the DATA step.
B. An incorrect variable name in the INPUT statement
C. A missing comma separating variables in the INPUT statement
D. A missing closing parenthesis in a function call.
E. A missing quotation mark in a character variable assignment.

**Answer: A**

Explanation:
The error message "ERROR 22-322: Syntax error, expecting a semicolon." is a common SAS error that indicates a missing semicolon at the end of a statement. This error usually occurs within the DATA step, as statements in the DATA step require a semicolon to terminate. While other options might cause errors, they would likely generate different error messages.

## Question: 8

You're working with a SAS program that includes a macro. The SAS log shows the following message:

What is the most likely reason for this warning, and how would you resolve it?

A. The macro variable &SIJM is not defined within the macro.
B. The macro is called before the variable SUM is created-
C. The macro is called with a missing parameter
D. The macro variable &SIJM is not used within the macro.
E. The macro variable &SIJM is used before it is defined in the current session.

**Answer: A**

Explanation:
The warning 'WARNING: Apparent symbolic reference &SUM not resolved" indicates that the macro is trying to use a variable named &SIJM, but it hasn't been defined within the scope of the macro. This means that the macro code is trying to access a variable that doesn't exist within the macro's context. To resolve it, you need to define the macro variable &SUM inside the macro's code. This can be done by assigning a value to it or passing it as a parameter when calling the macro.

## Question: 9

Your SAS program generates the following messages in the log:

```
NOTE: The data set WORK.DATA1 has 100 observations and 5 variables. NOTE: There were 0 observations read from the data set WORK.DATA1.
```

Which of the following statements are true about this situation? Select all that apply.

A. The program encountered an error during data input
B. The data set WORKDATAI is empty
C. The data set WORK.DATAI has data in it, but the program couldn't access it-
D. The program attempted to read from a data set named WORK. DATAI , but it doesn't exist.
E. There is a logical error in the code that prevents data from being read.

**Answer: B,E**

Explanation:
The SAS log messages indicate that a data set named WORKDATAI was created with 100 observations and 5 variables, but then zero observations were read from it. This suggests that the data set exists but is empty, or there's a problem in the program's logic preventing data from being read. Here's a breakdown: Option A (Incorrect): There's no indication of an error during data input. Option B (Correct): The first NOTE indicates the data set was created with observations, and the second NOTE confirms that no observations were read. This implies the data set is empty. Option C (Incorrect): If the data set had data but couldn't be accessed, there would likely be an error message. Option D (Incorrect): If the data set didn't exist, there wouldn't be a NOTE indicating the number of observations and variables. Option E (Correct): The most likely cause is a logical error in the code. The program might be trying to read from the wrong data set, skipping data due to a filter, or having an issue with the input statement.

## Question: 10

Which of the following SAS code snippets will result in a syntax error? Select all that apply.

A.

```
data work.test;  set sashelp.class;  if age > 10 then age = 10; run;
```

B.

```
data work.test;  set sashelp.class;  if age > 10 then age + 10; run;
```

C.

```
data work.test;  set sashelp.class;  if age > 10 then age=age+10; run;
```

D.

```
data work.test;  set sashelp.class;  if age > 10 then age=10;  run;
```

E.

```
data work.test;  set sashelp.class;  if age > 10 then age='10';  run;
```

**Answer: B,E**

Explanation:
Option B is incorrect as SAS does not support assigning a value to an expression. Option E is incorrect as the age variable is numeric, but the assignment is a character string. All other options are valid SAS syntax. Option A creates a new variable and assigns 10 if the age is greater than 10. Option C assigns the age value plus 10 to the age variable if the age is greater than 10. Option D is a valid SAS syntax with a conditional assignment.

# Thank You for Trying Our Product

**Special 16 USD Discount Coupon: NSZUBG3X**

**Email:** support@examsempire.com

**Check our Customer Testimonials and ratings available on every product page.**

**Visit our website.**

**https://examsempire.com/**