Print

Quiz, Lesson 9: Manipulating Data

Your Score: 100%

Congratulations! Your score of 100% indicates that you've mastered the topics in this lesson. If you'd like, you can review the feedback for each question.

✓

1. Based on this program and the observation shown in the PDV, what variable's value will be assigned to **Amount**?

```
data payroll;
   set salaries;
   if PayClass='Monthly' then Amount=Salary;
   else if PayClass='Hourly' then
      do;
         Amount=HrlyWage*Hrs;
         if Hrs>40 then Msg='CHECK TIMECARD';
      end;
   else Amount=JobRate;
run;
```

PDV

EmpID	PayClass	Hrs	Amount	JobRate	Msg
1201	Contract	30	.	.	

a. not specified
b. **HrlyWage*Hrs**
c. **JobRate**
d. **Salary**

Your answer: c
Correct answer: c

Because the value of **PayClass** is *Contract* and the value of **Hrs** is less than *40*, this observation doesn't meet either of the conditions for **PayClass**. Therefore, the ELSE statement with no condition assigns the **JobRate** value to **Amount**. The DO group doesn't apply in this case.

Review: Using DO Groups to Execute Multiple Statements

✓

2. Which of the following SAS functions returns a number from 1 to 12?

a. YEAR(*SAS-date*)
b. MONTH(*SAS-date*)
c. WEEKDAY(*SAS-date*)
d. TODAY(*SAS-date*)

Your answer: b
Correct answer: b

The MONTH function extracts the month from a SAS date and returns a number from 1 to 12.

Review: Using Date Functions in Assignment Statements

✓

3. An ELSE statement must immediately follow the IF-THEN statement in your program, and it executes only if the previous IF-THEN statement is true.

a. True
b. False

Your answer: b
Correct answer: b

The ELSE statement executes only if the previous IF-THEN statement is false. The ELSE statement specifies an alternative action to be performed when the condition in an IF-THEN statement is false.

Review: Using the ELSE Statement

✓

4. Which DATA step ensures that all observations are assigned a nonmissing value for **Bonus**?

a.

```
data work.bonus;
   set orion.sales;
   if Country='US' then Bonus=500;
   else if Country='AU' then Bonus=300;
run;
```

b.

```
data work.bonus;
   set orion.sales;
   if Country='US' then Bonus=500;
   else Bonus=300;
run;
```

Your answer: b
Correct answer: b

Adding an ELSE statement without a condition ensures that all observations have a nonmissing value for **Bonus**, even if they don't meet any stated condition.

Review: Using the ELSE Statement

✓

5. In the DATA step below, what is the length of the new variable, **Type**?

```
data orion.newloan;
   set orion.records;
   TotalPaid=sum(TotLoan+Interest);
   if Code='1' then Type='Fixed';
   else Type='Variable';
   length Type $ 10;
run;
```

a. 5
b. 8
c. 10
d. It depends on the first value of **Type** in **orion.records**.

Your answer: a
Correct answer: a

The length of a new variable is determined by the first reference in the DATA step. In this case, the length of **Type**, the new variable, is determined by the value *Fixed*, which is five characters long. Although a LENGTH statement is included, it is in the wrong place. It should appear before any other reference to the variable in the DATA step. The LENGTH statement cannot change the length of an existing variable.

Review: Avoiding Truncated Values When Creating Variables

✓

6. In the program below, what is the value of **Benefit** for the observation that is shown?

```
data work.total;
   set payroll.june;
   Benefit=sum(Ins,Health_Award);
run;
```

PDV

EmpID	Salary	Ins	Bonus	Health_Award	Benefit
KBA	5400	800	250	.	.

a. a missing value
b. *55050*
c. *800*
d. *0*

Your answer: c
Correct answer: c

The SUM function ignores missing values, so the value of **Benefit** is the sum of **Ins** (*800*) and **Health_Award** (a missing value), which equals *800*.

Review: Using the SUM Function in Assignment Statements

✓

7. Which of the following statements assigns the value *pass* to the variable **Grade** when the variable **Points** is greater than or equal to *70*?

a. `if Points=70 then Grade='pass';`
b. `if Grade>=70 then Points='pass';`
c. `if Points>=70 then Grade=pass;`
d. `if Points>=70 then Grade='pass';`

Your answer: d
Correct answer: d

The IF-THEN statement is a conditional statement; it executes a SAS statement for observations that meet specific conditions. In the IF-THEN statement, *expression* is a sequence of operands and operators that define a condition for selecting observations. After the THEN keyword, *statement* is any executable statement, such as the assignment statement. In this example, the expression is, "if the value of **Points** is greater than or equal to *70*," and the executable statement is, "then SAS assigns the value *pass* to the variable **Grade**." You enclose character values in quotation marks and you must specify them in the same case in which they appear in the data set.

Review: Using IF-THEN Statements

✓

8. Given what you know about how SAS processes the DROP and KEEP statements, would these two DATA steps create the same data set?

```
data work.subset1;
   set orion.sales;
   drop Salary;
   Bonus=500;
   Compensation=sum(Salary,Bonus);
   BonusMonth=month(Hire_Date);
run;
```

```
data work.subset1;
   set orion.sales;
   Bonus=500;
   Compensation=sum(Salary,Bonus);
   BonusMonth=month(Hire_Date);
   drop Salary;
run;
```

a. Yes

b. No

Your answer: a
Correct answer: a

Variables in the DROP statement are dropped during output, so they're available for calculations in the DATA step, even if they follow the statements that reference them.

Review: Creating Variables by Using Functions

✓

9. Which set of statements is equivalent to the code shown below?

```
if code='1' then Type='Fixed';
if code='2' then Type='Variable';
if code='1' then CheckRate='Yes';
if code ne'1' and code ne'2' then
   Type='Unknown';
```

a.

```
if code='1' then
   do;
      Type='Fixed';
      CheckRate='Yes';
   end;
   else if code='2' then
      Type='Variable';
   else Type='Unknown';
```

b.

```
if code='1' then Type='Fixed' and CheckRate='Yes';
if code='2' then Type='Variable';
else Type='Unknown';
```

c. both a and b

Your answer: a
Correct answer: a

The conditional statements in answer **a** assign the same values as the code shown, but they are more efficient. The first statement in answer **b** contains a syntax error. To execute multiple statements conditionally, you use a DO group.

Review: Using DO Groups to Execute Multiple Statements

✓

10. Use a DO group in a DATA step when you want to execute multiple statements for a true IF-THEN expression.

a. True

b. False

Your answer: **a**
Correct answer: **a**

Only one executable statement is allowed in IF-THEN and ELSE statements. Multiple executable statements are allowed in IF-THEN DO/ELSE DO statements. Each DO group can contain multiple statements that apply to the expression.

Review: Using DO Groups to Execute Multiple Statements

Close