Chapter 9: Manipulating Data

Chapter 9: Manipulating Data

Objectives

- Create data values using SAS functions.

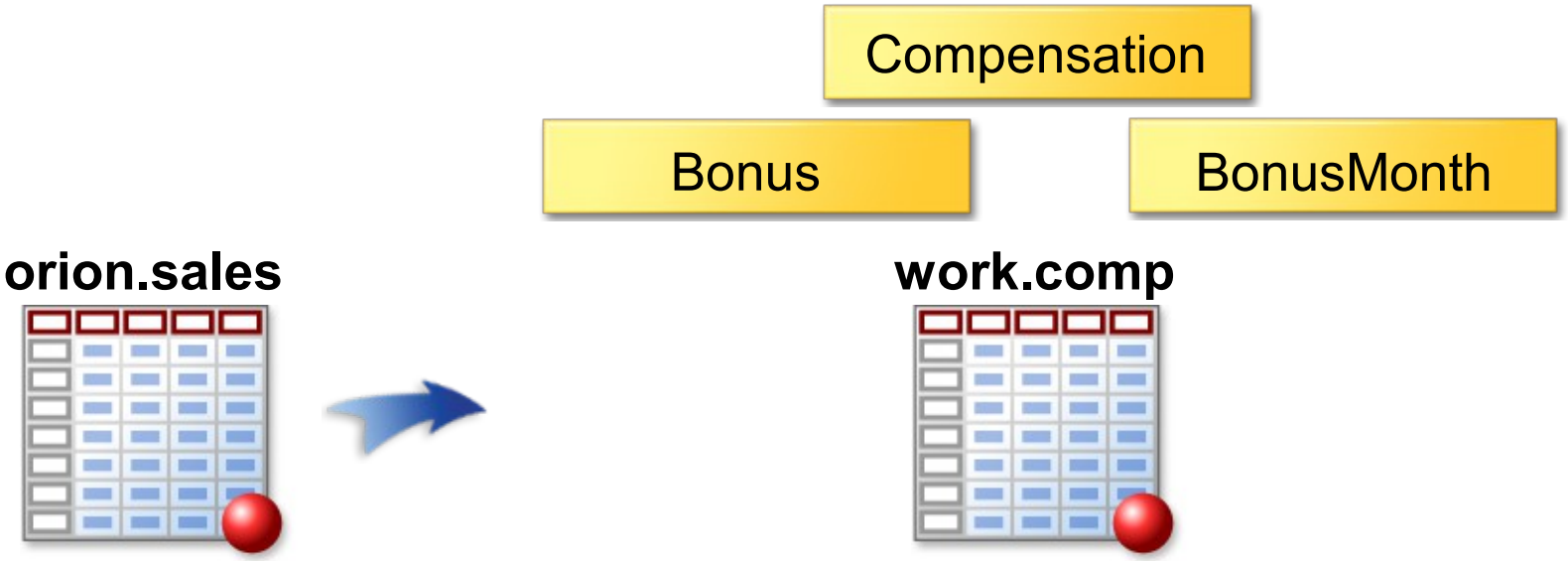
Business Scenario

Orion Star management plans to give a $500 bonus to each employee in his or her hire month.

Considerations

Create a new data set with three new variables:

- **Bonus**, which is a constant 500
- **Compensation**, which is the sum of **Salary** and **Bonus**
- **BonusMonth**, which is the month in which the employee was hired



Considerations

Partial **orion.sales**

Employee _ID	First _Name	Last _ Name	Gender	Salary	Job_ Title	Country	Birth_ Date	Hire_ Date
120102	Tom	Zhou	M	108255	Sales Manager	AU	3510	10744
120103	Wilson	Dawes	M	87975	Sales Manager	AU	-3996	5114
120121	Irenie	Elvish	F	26600	Sales Rep. II	AU	-5630	5114

Partial **work.comp**

Employee _ID	First _Name	Last _ Name	Bonus	Compensation	Bonus Month
120102	Tom	Zhou	500	108755	6
120103	Wilson	Dawes	500	88475	1
120121	Irenie	Elvish	500	27100	1

Drop **Gender**, **Salary**, **Job_Title**, **Country**, **Birth_Date**, and **Hire_Date** from **work.comp**.

9.01 Multiple Choice Poll

Which of the following statements creates a numeric variable **Bonus** with a value of 500?

- Bonus=$500;
- Bonus=500;
- label Bonus='500';
- format Bonus 500.;

Partial **work.comp**

Bonus	Compensation	Bonus Month
500	108755	6
500	88475	1
500	27100	1

9.01 Multiple Choice Poll – Correct Answer

Which of the following statements creates a numeric variable **Bonus** with a value of 500?

- Bonus=$500;
- Bonus=500;
- label Bonus='500';
- format Bonus 500.;

Partial **work.comp**

Bonus	Compensation	Bonus Month
500	108755	6
500	88475	1
500	27100	1

You use an assignment statement to set the value of the variable Bonus equal to 500. Numeric constants do not include commas or currency symbols.

SAS Functions

SAS functions can be used in an assignment statement. A *function* is a routine that accepts arguments and returns a value.

variable=*function-name*(*argument1*, *argument2*, …);

Some functions manipulate character values, compute descriptive statistics, or manipulate SAS date values.

- Arguments are enclosed in parentheses and separated by commas.
- A function can return a numeric or character result.

SUM Function

Use the *SUM function* to create **Compensation**. The SUM function is a descriptive statistics function that returns the sum of its arguments.

```
Compensation=sum(Salary,Bonus);
```

SUM(*argument1,argument2*, ...**)**

- The arguments must be numeric.
- Missing values are ignored by SUM and other descriptive statistics functions.

MONTH Function

Use the *MONTH function* to extract the month of hire from **Hire_Date**.

```
BonusMonth=month(Hire_Date);
```

MONTH(*SAS-date***)**

Other date functions can do the following:

- extract information from SAS date values
- create SAS date values

Date Functions: Extracting Values

Syntax	Description
YEAR(*SAS-date*)	Extracts the year from a SAS date and returns a four-digit year.
QTR(*SAS-date*)	Extracts the calendar quarter from a SAS date and returns a number from 1 to 4.
MONTH(*SAS-date*)	Extracts the month from a SAS date and returns a number from 1 to 12.
DAY(*SAS-date*)	Extracts the day of the month from a SAS date and returns a number from 1 to 31.
WEEKDAY(*SAS-date*)	Extracts the day of the week from a SAS date and returns a number from 1 to 7, where 1 represents Sunday.

Date Functions: Creating SAS Dates

Syntax	Description
TODAY() DATE()	Returns the current date as a SAS date value.
MDY(*month*,*day*,*year*)	Returns a SAS date value from numeric month, day, and year values.

Examples
CurrentDate=today();
y2k=mdy(01,1,2000);
NewYear=mdy(Mon,Day,2013);

Using SAS Functions

A function call can be used alone in an assignment statement.

```
BonusMonth=month(Hire_Date);
AnnivBonus=mdy(BonusMonth,15,2008);
```

A function call can be part of any SAS expression.

```
if month(Hire_Date)=12;
```

A function call can be an argument to another function.

```
AnnivBonus=mdy(month(Hire_Date),15,2012);
```

Using SAS Functions

Create **Bonus**, **Compensation**, and **BonusMonth**.

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

```
175  data work.comp;
176     set orion.sales;
177     Bonus=500;
178     Compensation=sum(Salary,Bonus);
179     BonusMonth=month(Hire_Date);
180  run;

NOTE: There were 165 observations read from the data set ORION.SALES.
NOTE: The data set WORK.COMP has 165 observations and 12 variables.
```

orion.sales has nine variables.

p109d01

Viewing the Output

```
proc print data=work.comp noobs;
   var Employee_ID First_Name Last_Name
       Bonus Compensation BonusMonth;
run;
```

Partial PROC PRINT Output

Employee_ID	First_ Name	Last_Name	Bonus	Bonus Compensation	Month
120102	Tom	Zhou	500	108755	6
120103	Wilson	Dawes	500	88475	1
120121	Irenie	Elvish	500	27100	1
120122	Christina	Ngan	500	27975	7
120123	Kimiko	Hotstone	500	26690	10

p109d01

9.02 Quiz

A DROP statement has been added to this DATA step. Will the program calculate **Compensation** and **BonusMonth** correctly?

```
data work.comp;
   set orion.sales;
   drop Gender Salary Job_Title Country
        Birth_Date Hire_Date;
   Bonus=500;
   Compensation=sum(Salary,Bonus);
   BonusMonth=month(Hire_Date);
run;
```

p109a01

9.02 Quiz – Correct Answer

A DROP statement has been added to this DATA step. Will the program calculate **Compensation** and **BonusMonth** correctly?

```
23   data work.comp;
24      set orion.sales;
25      drop Gender Salary Job_Title Country
26              Birth_Date Hire_Date;
27      Bonus=500;
28      Compensation=sum(Salary,Bonus);
29      BonusMonth=month(Hire_Date);
30   run;
NOTE: There were 165 observations read from the data set ORION.SALES.
NOTE: The data set WORK.COMP has 165 observations and 6 variables.
```

Yes. A drop flag is set for the dropped variables, but the variables are in the PDV and therefore available for processing. DROP is a compile-time only statement.

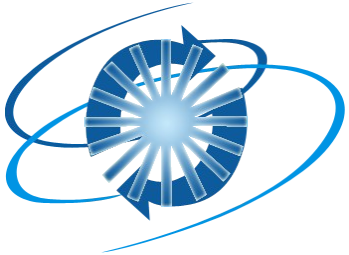
Viewing the Output

```
proc print data=work.comp noobs;
run;
```

Partial PROC PRINT Output

Employee_ID	First_ Name	Last_Name	Bonus	Compensation	Month
120102	Tom	Zhou	500	108755	6
120103	Wilson	Dawes	500	88475	1
120121	Irenie	Elvish	500	27100	1
120122	Christina	Ngan	500	27975	7
120123	Kimiko	Hotstone	500	26690	10

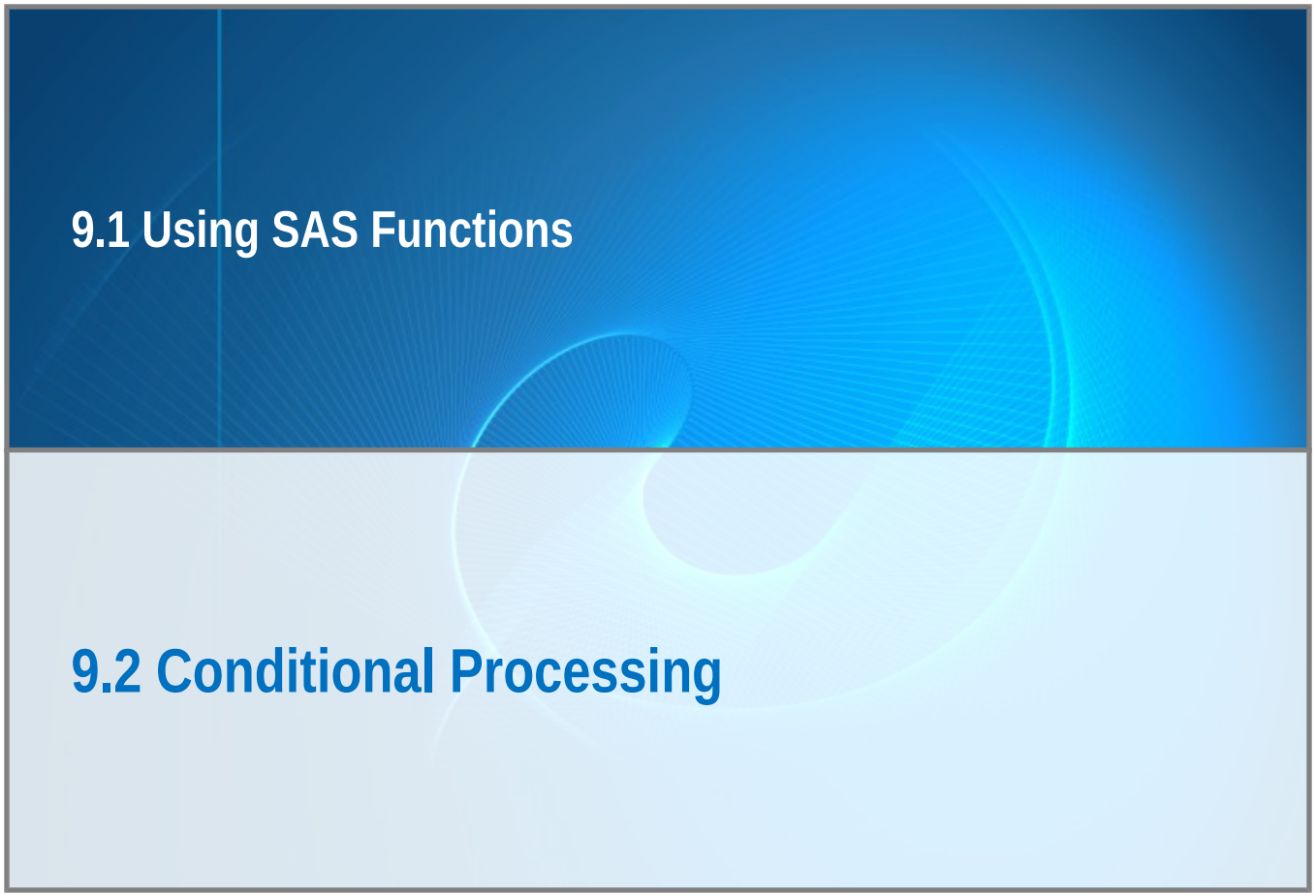
p109a01



Exercise

This exercise reinforces the concepts discussed previously.

Chapter 9: Manipulating Data



Objectives

- Process data conditionally using IF-THEN/ELSE statements.
- Execute multiple statements conditionally using DO and END statements.
- Control the length of character variables using the LENGTH statement.

Business Scenario

Orion Star management plans to give each sales employee a bonus based on his or her job title.

Considerations

Create a new data set, **work.comp**, using **orion.sales** as input. Include a new variable, **Bonus**, with a value based on **Job_Title**.

Job_Title	Bonus
Sales Rep. IV	1000
Sales Manager	1500
Senior Sales Manager	2000
Chief Sales Officer	2500

IF-THEN Statements

The IF-THEN statement executes a SAS statement for observations that meet a specific condition.

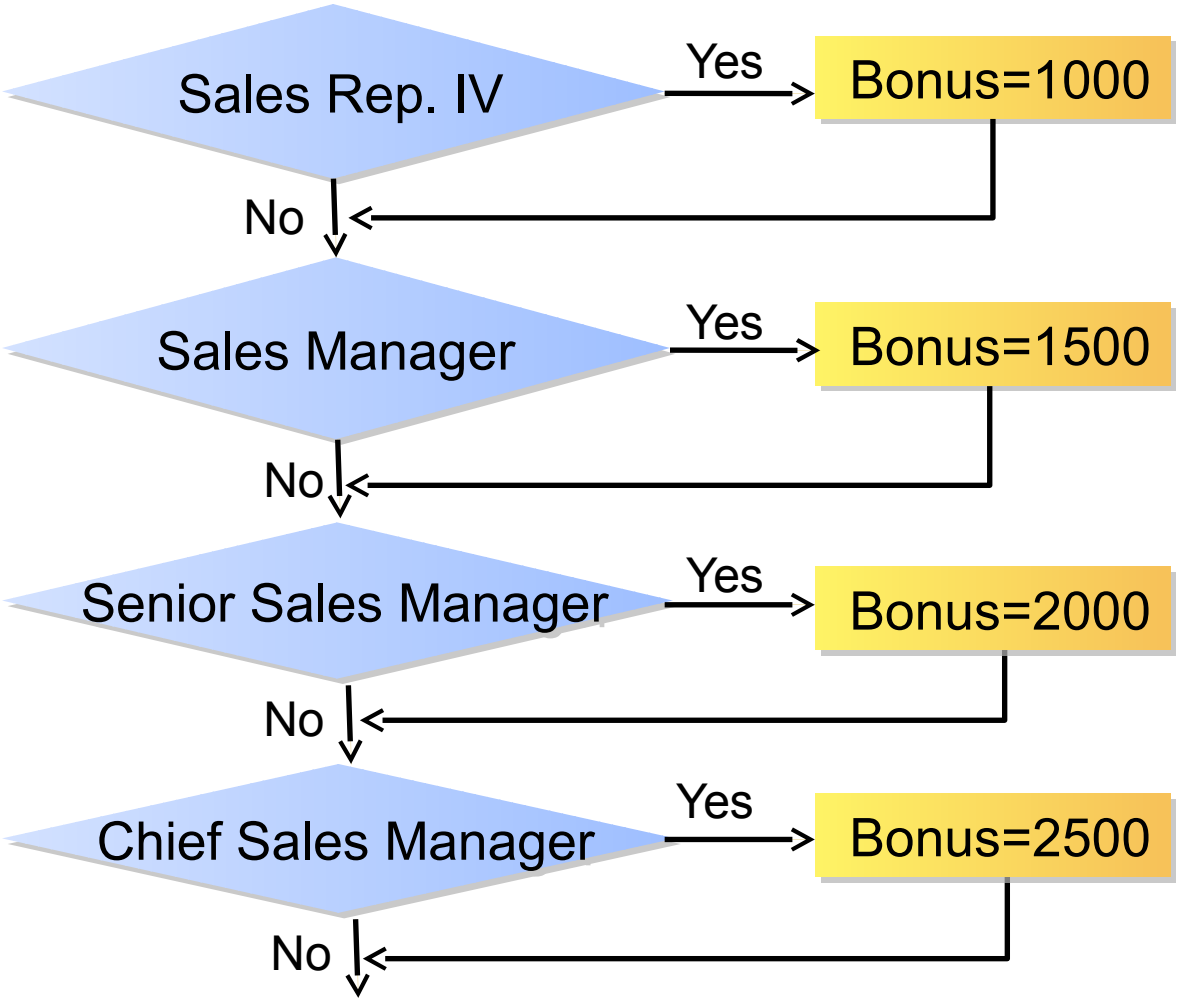
```
data work.comp;
   set orion.sales;
   if Job_Title='Sales Rep. IV' then
                        Bonus=1000;
   ...
run;
```

IF *expression* **THEN** *statement*;

- *expression* defines a condition.
- *statement* can be any executable SAS statement.
- If *expression* is true, then *statement* executes.

Conditional Processing

The value assigned to **Bonus** is determined by testing for various values of **Job_Title**.



Conditional Processing

```
data work.comp;
    set orion.sales;
    if Job_Title='Sales Rep. IV' then
       Bonus=1000;
    if Job_Title='Sales Manager' then
       Bonus=1500;
    if Job_Title='Senior Sales Manager'
       then Bonus=2000;
    if Job_Title='Chief Sales Officer'
       then Bonus=2500;
run;
```

PDV

Employee_ID	Last_Name	⋯	Job_Title	Bonus
120102	Zhou	⋯	Sales Manager	.

p109d02

Conditional Processing

```
data work.comp;
    set orion.sales;
    if Job_Title='Sales Rep. IV' then
       Bonus=1000;
    if Job_Title='Sales Manager' then
       Bonus=1500;
    if Job_Title='Senior Sales Manager'
       then Bonus=2000;
    if Job_Title='Chief Sales Officer'
       then Bonus=2500;
run;
```

false

PDV

Employee_ID	Last_Name	···	Job_Title	Bonus
120102	Zhou		Sales Manager	.

...

Conditional Processing

```
data work.comp;
    set orion.sales;
    if Job_Title='Sales R                V' then
       Bonus=1000;
    if Job_Title='Sales Manager' then
       Bonus=1500;
    if Job_Title='Senior Sales Manager'
       then Bonus=2000;
    if Job_Title='Chief Sales Officer'
       then Bonus=2500;
run;
```

true

PDV

Employee_ID	Last_Name	···	Job_Title	Bonus
120102	Zhou		Sales Manager	.

...

Conditional Processing

```
data work.comp;
    set orion.sales;
    if Job_Title='Sales Rep. IV' then
       Bonus=1000;
    if Job_Title='Sales Manager' then
       Bonus=1500;
    if Job_Title='Senior Sales Manager'
       then Bonus=2000;
    if Job_Title='Chief Sales Officer'
       then Bonus=2500;
run;
```

PDV

Employee_ID	Last_Name	···	Job_Title	Bonus
120102	Zhou		Sales Manager	1500

...

Conditional Processing

```
data work.comp;
    set orion.sales;
    if Job_Title='Sales Rep. IV' then
       Bonus=1000;
    if Job_Title='Sales M          r' then
       Bonus=1500;
    if Job_Title='Senior Sales Manager'
       then Bonus=2000;
    if Job_Title='Chief Sales Officer'
       then Bonus=2500;
run;
```

false

PDV

Employee_ID	Last_Name	···	Job_Title	Bonus
120102	Zhou		Sales Manager	1500

...

Conditional Processing

```
data work.comp;
    set orion.sales;
    if Job_Title='Sales Rep. IV' then
       Bonus=1000;
    if Job_Title='Sales Manager' then
       Bonus=1500;
    if Job_Title='Senior                Manager'
       then Bonus=2000;
    if Job_Title='Chief Sales Officer'
       then Bonus=2500;
run;
```

false

PDV

Employee_ID	Last_Name	···	Job_Title	Bonus
120102	Zhou		Sales Manager	1500

...

Conditional Processing

```
data work.comp;
    set orion.sales;
    if Job_Title='Sales Rep. IV' then
       Bonus=1000;
    if Job_Title='Sales Manager' then
       Bonus=1500;
    if Job_Title='Senior Sales Manager'
       then Bonus=2000;
    if Job_Title='Chief Sales Officer'
       then Bonus=2500;
run;
```

Implicit OUTPUT;
Implicit RETURN;

PDV

Employee_ID	Last_Name	⋯	Job_Title	Bonus
120102	Zhou	⋯	Sales Manager	1500

Conditional Processing

```
data work.comp;
    se                    s;
    if Job_Title='Sales Rep. IV' then
       Bonus=1000;
    if Job_Title='Sales Manager' then
       Bonus=1500;
    if Job_Title='Senior Sales Manager'
       then Bonus=2000;
    if Job_Title='Chief Sales Officer'
       then Bonus=2500;
run;
```

Continue until EOF

PDV

Employee_ID	Last_Name	···	Job_Title	Bonus
120102	Zhou		Sales Manager	1500

Viewing the Output

```
proc print data=work.comp;
   var Last_Name Job_Title Bonus;
run;
```

Partial PROC PRINT Output

Obs	Last_Name	Job_Title	Bonus
1	Zhou	Sales Manager	1500
2	Dawes	Sales Manager	1500
3	Elvish	Sales Rep. II	.
4	Ngan	Sales Rep. II	.
5	Hotstone	Sales Rep. I	.
6	Daymond	Sales Rep. I	.
7	Hofmeister	Sales Rep. IV	1000
8	Denny	Sales Rep. II	.
9	Clarkson	Sales Rep. II	.
10	Kletschkus	Sales Rep. IV	1000
11	Roebuck	Sales Rep. III	.
12	Lyon	Sales Rep. I	.

p109d02

9.03 Multiple Choice Poll

In the previous program, is it possible for more than one condition to be true for a single observation?

- Yes, more than one condition can be true.
- No, the conditions are mutually exclusive, so only one condition can be true.

9.03 Multiple Choice Poll – Correct Answer

In the previous program, is it possible for more than one condition to be true for a single observation?

- Yes, more than one condition can be true.
- No, the conditions are mutually exclusive, so only one condition can be true.

For each observation there is only one value for Job_Title. If that value matches one of the conditions, then it cannot match any other condition.

Using the ELSE Statement

Use the *ELSE statement* when testing mutually exclusive conditions.

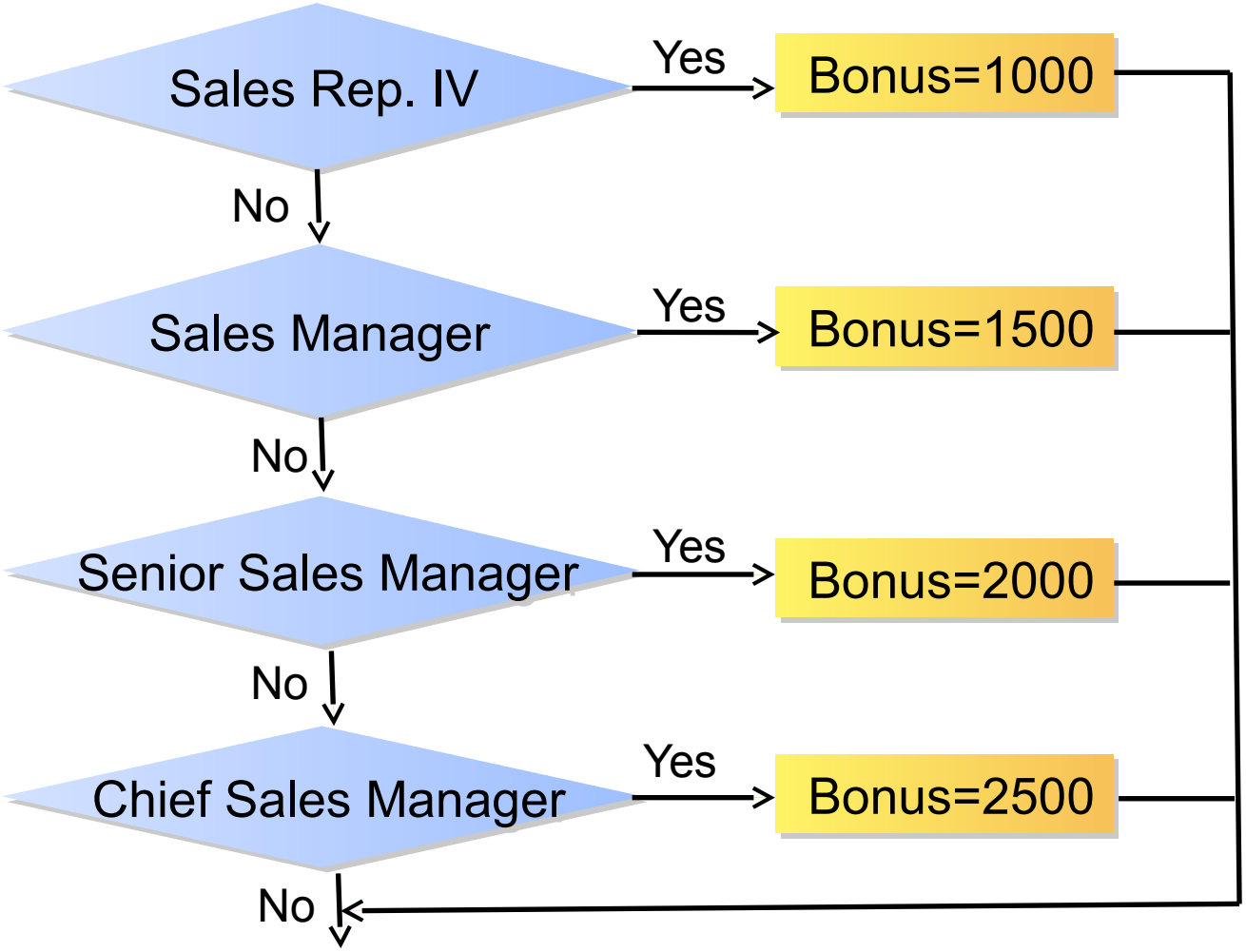
```
data work.comp;
   set orion.sales;
   if Job_Title='Sales Rep. IV' then
      Bonus=1000;
   else if Job_Title='Sales Manager' then
Bonus=1500;
   else if Job_Title='Senior Sales Manager'
    then Bonus=2000;
   else if Job_Title='Chief Sales Officer'
    then Bonus=2500;
run;
```

IF *expression* **THEN** *statement*;
<ELSE IF *expression* **THEN** *statement*;**>**
<ELSE IF *expression* **THEN** *statement*;**>**

p109d03

Conditional Processing

When an expression is true, the associated statement is executed and subsequent ELSE statements are skipped.



IF-THEN Statements

```
data work.comp;
    set orion.sales;
    if Job_Title='Sales Rep. IV' then
     Bonus=1000;
    else if Job_Title='Sales Manager' then
Bonus=1500;
    else if Job_Title='Senior Sales Manager'
     then Bonus=2000;
    else if Job_Title='Chief Sales Officer'
     then Bonus=2500;
run;
```

PDV

Employee_ID	Last_Name	…	Job_Title	Bonus
120102	Zhou	…	Sales Manager	.

p109d03

IF-THEN Statements

```
data work.comp;
   set orion.sales;
   if Job_Title='Sales Rep. IV' then
    Bonus=1000;
   else if Job_Title='Sales Manager' then
Bonus=1500;
   else if Job_Title='Senior Sales Manager'
    then Bonus=2000;
   else if Job_Title='Chief Sales Officer'
    then Bonus=2500;
run;
```

false

PDV

Employee_ID	Last_Name	…	Job_Title	Bonus
120102	Zhou	…	Sales Manager	.

…

IF-THEN Statements

```
data work.comp;
   set orion.sales;
   if Job_Title='Sales Rep. I     en
    Bonus=1000;
   else if Job_Title='Sales Manager' then
Bonus=1500;
   else if Job_Title='Senior Sales Manager'
    then Bonus=2000;
   else if Job_Title='Chief Sales Officer'
    then Bonus=2500;
run;
```

true

PDV

Employee_ID	Last_Name	…	Job_Title	Bonus
120102	Zhou	…	Sales Manager	.

…

IF-THEN Statements

```
data work.comp;
    set orion.sales;
    if Job_Title='Sales Rep. IV' then
     Bonus=1000;
    else if Job_Title='Sales Manager' then
Bonus=1500;
    else if Job_Title='Senior Sales Manager'
     then Bonus=2000;
    else if Job_Title='Chief Sales Officer'
     then Bonus=2500;
run;
```

PDV

Employee_ID	Last_Name	⋯	Job_Title	Bonus
120102	Zhou	⋯	Sales Manager	1500

...

IF-THEN Statements

```
data work.comp;
   set orion.sales;
   if Job_Title='Sales Rep. IV' then
      Bonus=1000;
   else if Job_Title='Sales Manager' then
Bonus=1500;
   else if Job_Title='Senior Sales Manager'
    then Bonus=2000;
   else if Job_Title='Chief Sales Officer'
    then Bonus=2500;
run;
```

Implicit OUTPUT;
Implicit RETURN;

PDV

Employee_ID	Last_Name	…	Job_Title	Bonus
120102	Zhou	…	Sales Manager	1500

IF-THEN Statements

```
data work.comp;
   set orion.sales;
   if Job_Title='Sales Rep. IV' then
      Bonus=1000;
   else if Job_Title='Sales Manager' then
Bonus=1500;
   else if Job_Title='Senior Sales Manager'
    then Bonus=2000;
   else if Job_Title='Chief Sales Officer'
    then Bonus=2500;
run;
```

Continue until EOF

PDV

Employee_ID	Last_Name	⋯	Job_Title	Bonus
120102	Zhou		Sales Manager	1500

Viewing the Output

```
proc print data=work.comp;
   var Last_Name Job_Title Bonus;
run;
```

Partial PROC PRINT Output

Obs	Last_Name	Job_Title	Bonus
1	Zhou	Sales Manager	1500
2	Dawes	Sales Manager	1500
3	Elvish	Sales Rep. II	.
4	Ngan	Sales Rep. II	.
5	Hotstone	Sales Rep. I	.
6	Daymond	Sales Rep. I	.
7	Hofmeister	Sales Rep. IV	1000
8	Denny	Sales Rep. II	.
9	Clarkson	Sales Rep. II	.
10	Kletschkus	Sales Rep. IV	1000

p109d03

Business Scenario: Part 2

Orion Star management wants to modify the bonus plan as defined below.

Job_Title	Bonus
Sales Rep. III	1000
Sales Rep. IV	1000
Sales Manager	1500
Senior Sales Manager	2000
Chief Sales Officer	2500
All other titles	500

Using Conditional Processing

```
data work.comp;
   set orion.sales;
   if Job_Title='Sales Rep. III' or
      Job_Title='Sales Rep. IV' then
       Bonus=1000;
   else if Job_Title='Sales Manager' then
      Bonus=1500;
   else if Job_Title='Senior Sales Manager'
      then Bonus=2000;
   else if Job_Title='Chief Sales Officer'
      then Bonus=2500;
   else Bonus=500;
run;
```

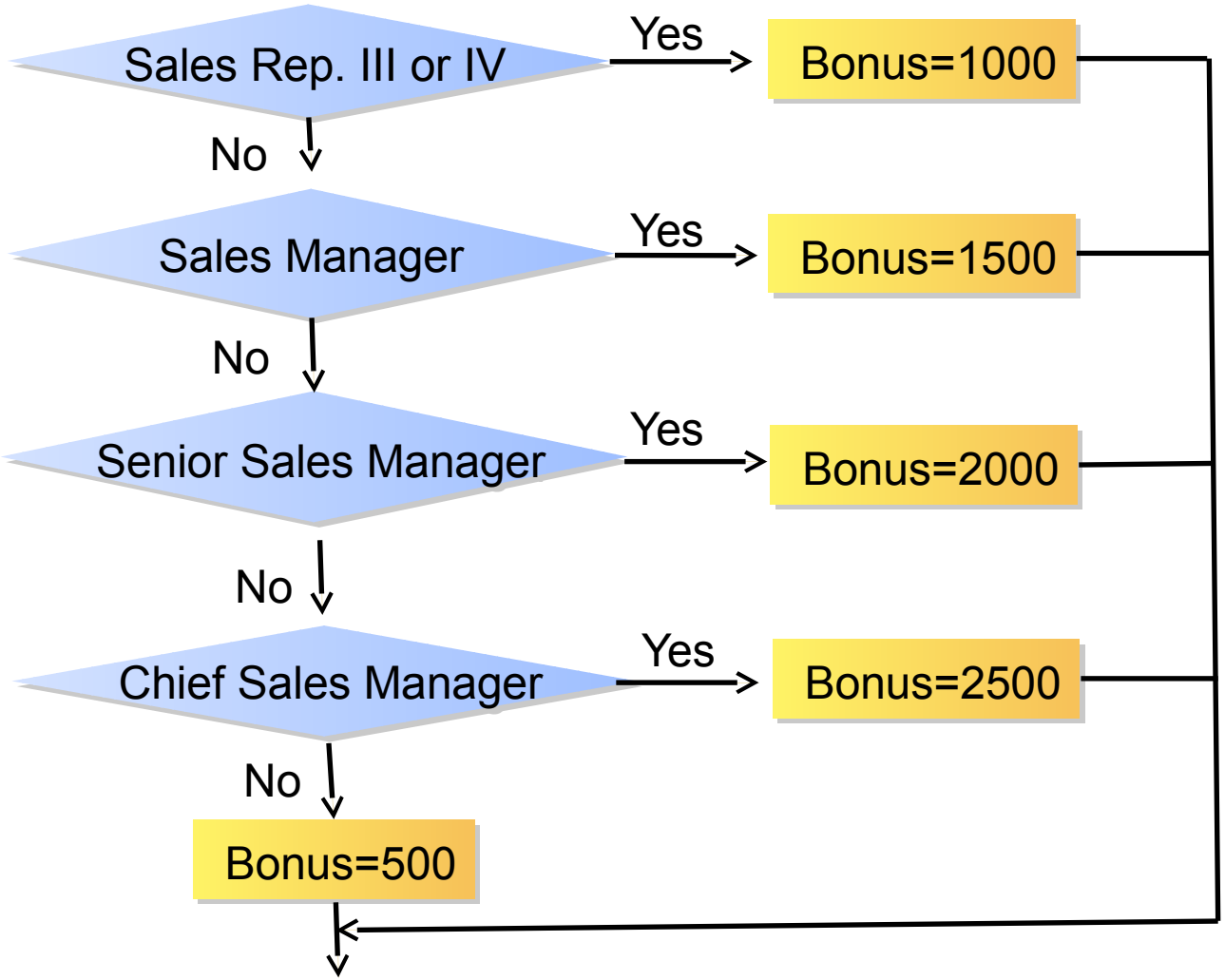
compound condition

IF *expression* **THEN** *statement*;
<ELSE IF *expression* **THEN** *statement*;**>**
<…>
<**ELSE** *statement*;>

p109d04

Conditional Processing

An optional final ELSE statement gives an alternative action if none of the conditions are true.



Viewing the Output

```
proc print data=work.comp;
   var Last_Name Job_Title Bonus;
run;
```

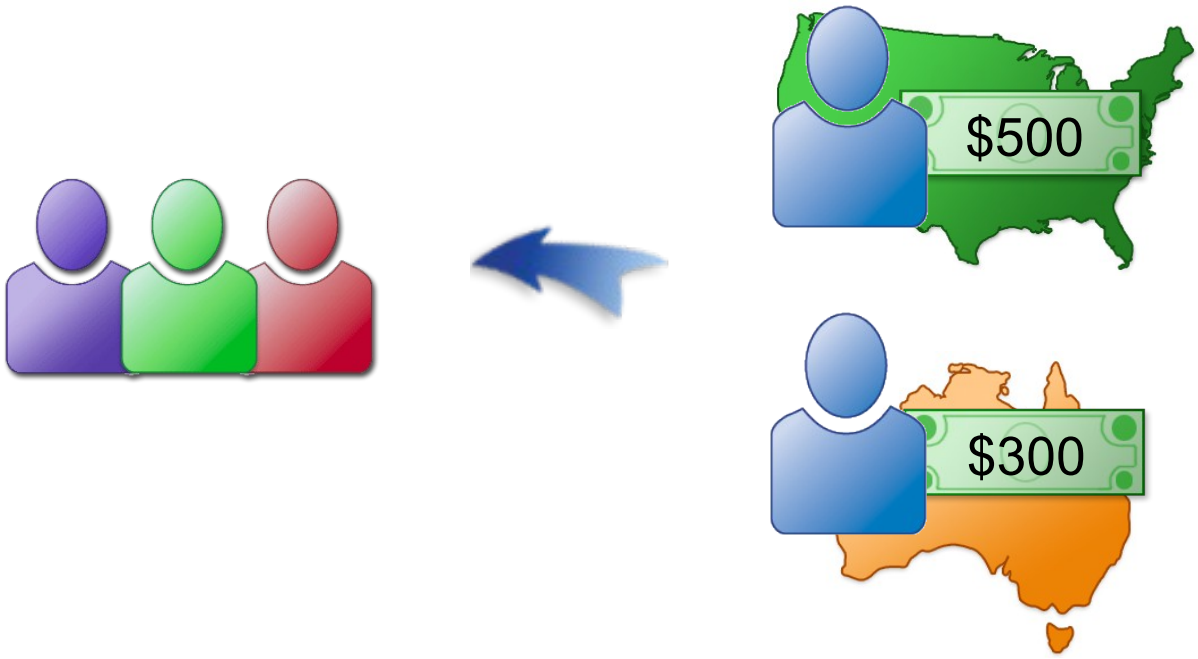
Partial PROC PRINT Output

Obs	Last_Name	Job_Title	Bonus
1	Zhou	Sales Manager	1500
2	Dawes	Sales Manager	1500
3	Elvish	Sales Rep. II	500
4	Ngan	Sales Rep. II	500
5	Hotstone	Sales Rep. I	500
6	Daymond	Sales Rep. I	500
7	Hofmeister	Sales Rep. IV	1000
8	Denny	Sales Rep. II	500
9	Clarkson	Sales Rep. II	500
10	Kletschkus	Sales Rep. IV	1000

p109d04

Business Scenario

Orion Star managers are considering a country-based bonus. Create a new SAS data set named **work.bonus** using **orion.sales** as input. The value of the new variable, **Bonus**, is based on **Country**.



IF-THEN/ELSE Statements

If **orion.sales** has been validated and ***only*** includes the **Country** values *US* and *AU*, the conditional clause can be omitted from the ELSE statement.

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

IF *expression* **THEN** *statement*;
ELSE *statement*;

All observations not equal to *US* get a bonus of 300.

p109d05

Viewing the Output

```
proc print data=work.bonus;
   var First_Name Last_Name Country Bonus;
run;
```

Partial PROC PRINT Output

Obs	First_Name	Last_Name	Country	Bonus
60	Billy	Plested	AU	300
61	Matsuoka	Wills	AU	300
62	Vino	George	AU	300
63	Meera	Body	AU	300
64	Harry	Highpoint	US	500
65	Julienne	Magolan	US	500
66	Scott	Desanctis	US	500
67	Cherda	Ridley	US	500
68	Priscilla	Farren	US	500
69	Robert	Stevens	US	500

9.04 Quiz

Program **p109a02** reads **orion.nonsales**, a non-validated data set. Open and submit the program and review the results. Why is **Bonus** set to 300 in observations 125, 197, and 200?

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

9.04 Quiz – Correct Answer

Program **p109a02** reads **orion.nonsales**, a non-validated data set. Open and submit the program and review the results. Why is **Bonus** set to 300 in observations 125, 197, and 200?

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

The Country variable has some mixed case values in orion.nonsales. Observations with a country value of *US* are assigned 500. All others are assigned 300, including *us*.

Testing for Invalid Data

You can test for multiple values of **Country**.

```
data work.bonus;
   set orion.nonsales;
   if Country in ('US','us')
      then Bonus=500;
   else Bonus=300;
run;
```

You can use the UPCASE function in the expression.

```
data work.bonus;
   set orion.nonsales;
   if upcase(Country)='US'
      then Bonus=500;
   else Bonus=300;
run;
```

p109a02s

Cleaning Invalid Data

You can clean the data before checking the value.

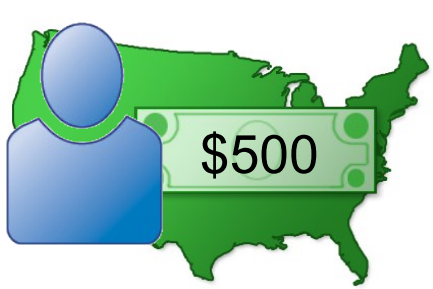
```
data work.bonus;
   set orion.nonsales;
   Country=upcase(Country);
   if Country='US'
      then Bonus=500;
   else Bonus=300;
run;
```

- It is a best practice to clean the data at the source, but in some cases, that is not possible. With this method, you are creating a clean data set.

Business Scenario

Orion Star employees will receive a bonus once or twice a year. In addition to **Bonus**, add a new variable, **Freq**, that is equal to the following:

- *Once a Year* for United States employees
- *Twice a Year* for Australian employees



Once a Year

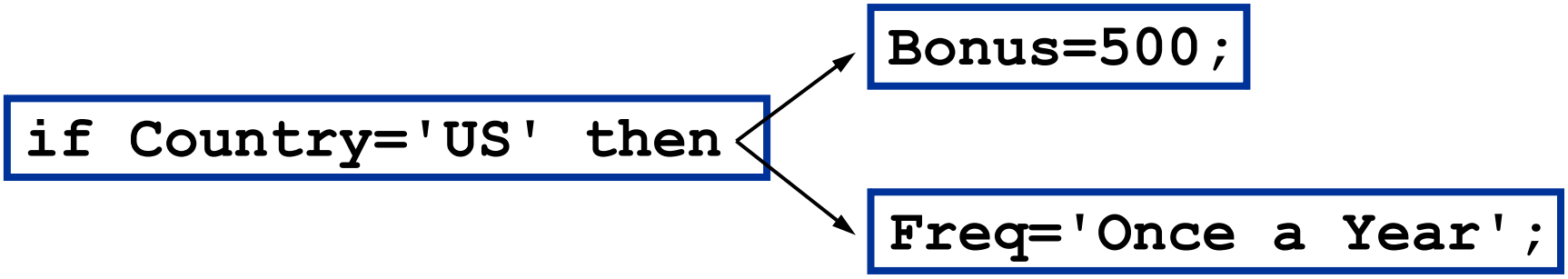
Twice a Year

IF-THEN/ELSE Statements

Only ***one*** executable statement is allowed in IF-THEN and ELSE statements.

IF *expression* **THEN** *statement*;
ELSE IF *expression* **THEN** *statement*;
ELSE *statement*;

For this business scenario, ***two*** statements must be executed for each true expression.



DO Group

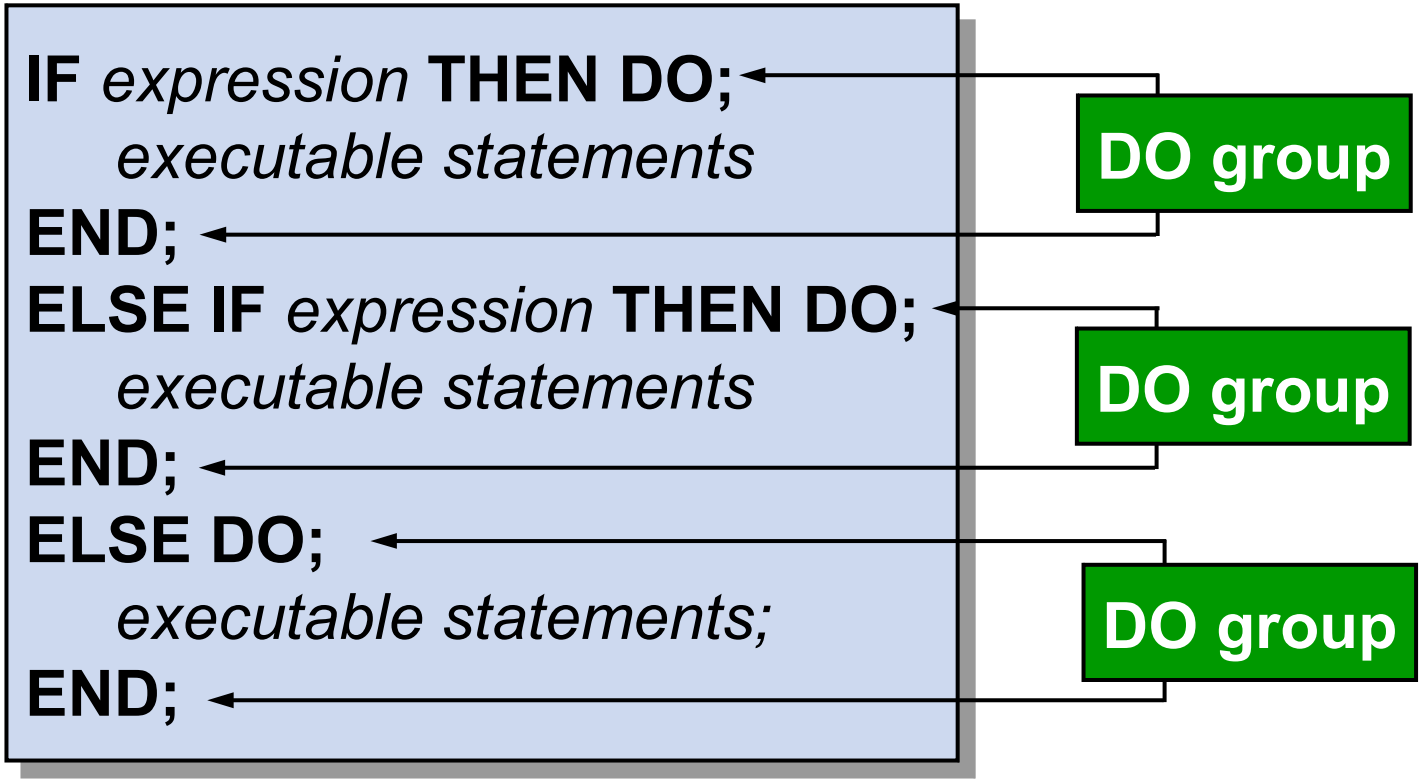
Multiple statements are permitted in a *DO group*.

```
data work.bonus;
    set orion.sales;
    if Country='US' then do;
       Bonus=500;
     Freq='Once a Year';
    end;
    else if Country='AU' then do;
       Bonus=300;
     Freq='Twice a Year';
    end;
run;
```

Each DO group ends with an END statement.

IF-THEN DO/ELSE DO Statements

Multiple statements are also permitted in an ELSE DO group.



Viewing the Output

```
proc print data=work.bonus;
    var First_Name Last_Name Country Bonus
        Freq;
run;
```

Partial PROC PRINT Output

Obs	First_Name	Last_Name	Country	Bonus	Freq
60	Billy	Plested	AU	300	Twice a Yea
61	Matsuoka	Wills	AU	300	Twice a Yea
62	Vino	George	AU	300	Twice a Yea
63	Meera	Body	AU	300	Twice a Yea
64	Harry	Highpoint	US	500	Once a Year
65	Julienne	Magolan	US	500	Once a Year
66	Scott	Desanctis	US	500	Once a Year
67	Cherda	Ridley	US	500	Once a Year
68	Priscilla	Farren	US	500	Once a Year
69	Robert	Stevens	US	500	Once a Year

truncation

p109d07

Compilation

```
data work.bonus;
   set orion.sales;
   if Country='US' then do;
      Bonus=500;
    Freq='Once a Year';
   end;
   else if Country='AU' then do;
      Bonus=300;
    Freq='Twice a Year';
   end;
run;
```

PDV

Employee_ID N 8	First_Name $ 12	...	Hire_Date N 8

p109d07

Compilation

```
data work.bonus;
   set orion.sales;
   if Country='US' then do;
      Bonus=500;
    Freq='Once a Year';
   end;
   else if Country='AU' then do;
      Bonus=300;
    Freq='Twice a Year';
   end;
run;
```

PDV

Employee_ID N 8	First_Name $ 12	...	Hire_Date N 8	Bonus N 8

...

Compilation

```
data work.bonus;
    set orion.sales;
    if Country='US' then do;
       Bonus=500;
     Freq='Once a Year';
    end;
    else if (                    then do;
       Bonus=300;
     Freq='Twice a Year';
    end;
run;
```

11 characters

PDV

Employee_ID N 8	First_Name $ 12	...	Hire_Date N 8	Bonus N 8	Freq $ 11

...

Compilation

```
data work.bonus;
   set orion.sales;
   if Country='US' then do;
      Bonus=500;
    Freq='Once a Year';
   end;
   else if Country='AU' then do;
      Bonus=300;
    Freq='Twice a Year';
   end;
run;
```

12 characters

length does not change

PDV

Employee_ID N 8	First_Name $ 12	...	Hire_Date N 8	Bonus N 8	Freq $ 11

9.05 Quiz

How would you prevent **Freq** from being truncated?

9.05 Quiz – Correct Answer

How would you prevent **Freq** from being truncated?

Possible solutions:

- **Pad the first occurrence of the Freq value with blanks to be the length of the longest possible value.**
- **Switch conditional statements to place the longest value of Freq in the first conditional statement.**
- **Add a LENGTH statement to declare the byte size of the variable up front.**

Defining Character Variables

Set the length of the variable **Freq** to avoid truncation.

```
data work.bonus;
    set orion.sales;
    length Freq $ 12;
    if Country='US' then do;
       Bonus=500;
     Freq='Once a Year';
    end;
    else if Country='AU' then do;
       Bonus=300;
     Freq='Twice a Year';
    end;
run;
```

LENGTH *variable(s)* <$> *length*;

- It is a good practice to use a LENGTH statement any time you create a new character variable.

p109d08

Compilation

```
data work.bonus;
    set orion.sales;
    length Freq $ 12;
    if Country='US' then do;
       Bonus=500;
     Freq='Once a Year';
    end;
    else if Country='AU' then do;
       Bonus=300;
     Freq='Twice a Year';
    end;
run;
```

PDV

Employee_ID N 8	First_Name $ 12	...	Hire_Date N 8

p109d08

Compilation

```
data work.bonus;
    set orion.sales;
    length Freq $ 12;
    if Country='US' then do;
       Bonus=500;
     Freq='Once a Year';
    end;
    else if Country='AU' then do;
       Bonus=300;
     Freq='Twice a Year';
    end;
run;
```

PDV

Employee_ID N 8	First_Name $ 12	...	Hire_Date N 8	Freq $ 12

...

Compilation

```
data work.bonus;
    set orion.sales;
    length Freq $ 12;
    if Country='US' then do;
       Bonus=500;
     Freq='Once a Year';
    end;
    else if Country='AU' then do;
       Bonus=300;
     Freq='Twice a Year';
    end;
run;
```

PDV

Employee_ID N 8	First_Name $ 12	...	Hire_Date N 8	Freq $ 12	Bonus N 8

...

Compilation

```
data work.bonus;
   set orion.sales;
   length Freq $ 12;
   if Country='US' then do;
      Bonus=500;
    Freq='Once a Year';
   end;
   else if Country='AU' then do;
      Bonus=300;
    Freq='Twice a Year';
   end;
run;
```

length does not change

PDV

Employee_ID N 8	First_Name $ 12	...	Hire_Date N 8	Freq $ 12	Bonus N 8

Viewing the Output

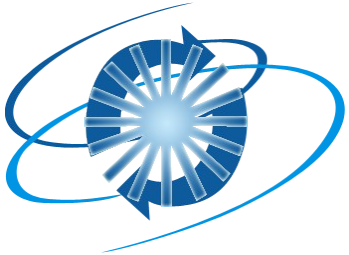
```
proc print data=work.bonus;
   var First_Name Last_Name Country
       Bonus Freq;
run;
```

Partial PROC PRINT Output

Obs	First_Name	Last_Name	Country	Bonus	Freq
60	Billy	Plested	AU	300	Twice a Year
61	Matsuoka	Wills	AU	300	Twice a Year
62	Vino	George	AU	300	Twice a Year
63	Meera	Body	AU	300	Twice a Year
64	Harry	Highpoint	US	500	Once a Year
65	Julienne	Magolan	US	500	Once a Year
66	Scott	Desanctis	US	500	Once a Year
67	Cherda	Ridley	US	500	Once a Year
68	Priscilla	Farren	US	500	Once a Year
69	Robert	Stevens	US	500	Once a Year

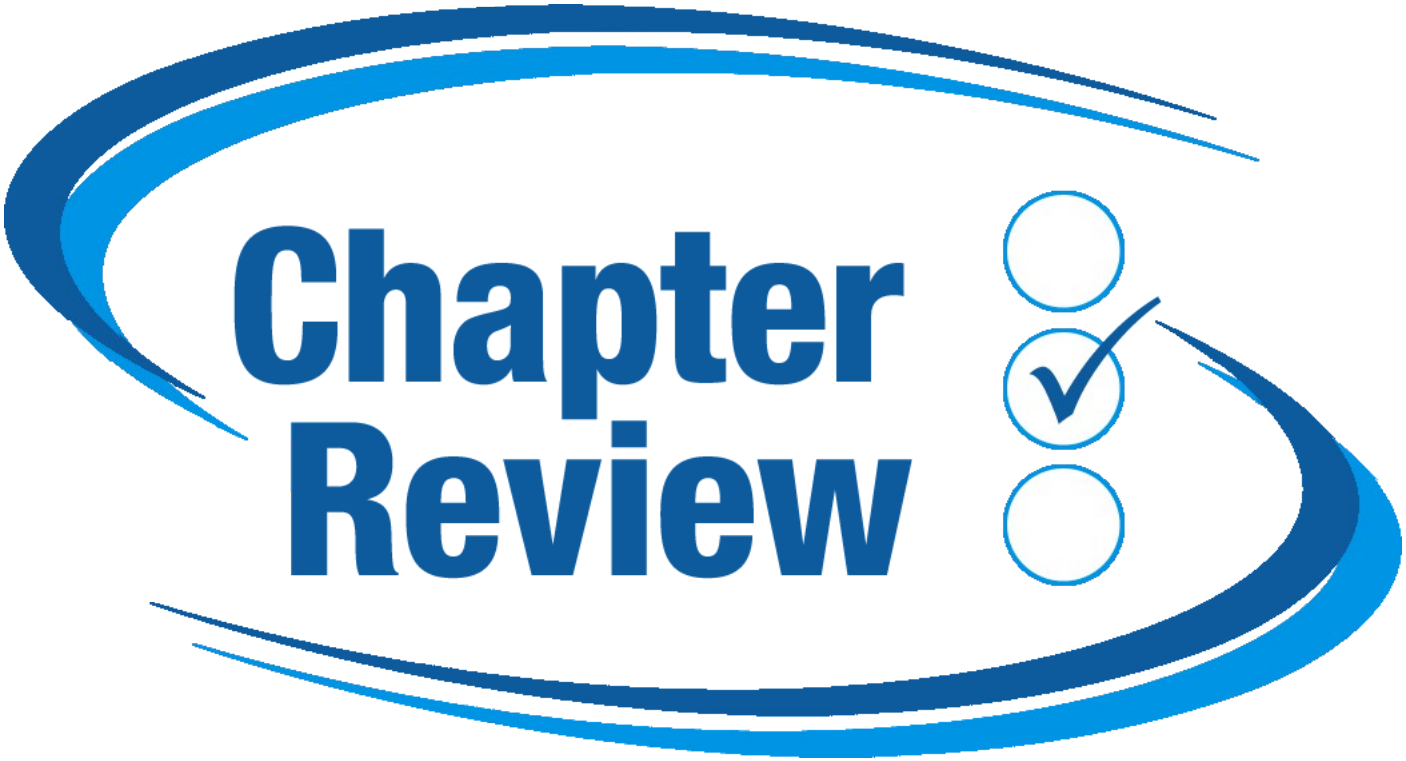
no truncation

p109d08



Exercise

This exercise reinforces the concepts discussed previously.



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;
```

- not specified
- HrlyWage*Hrs
- JobRate
- Salary

PDV

Emp ID	Pay Class	Hrs	Amount	Job Rate	Msg
1201	Contract	30	.	.	

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

Emp ID	Pay Class	Hrs	Amount	Job Rate	Msg
1201	Contract	30	.	.	

- not specified
- HrlyWage*Hrs
- JobRate
- Salary

2. Which of the following SAS functions returns a number from 1 to 12?
   - YEAR(*SAS-date*)
   - MONTH(*SAS-date*)
   - WEEKDAY(*SAS-date*)
   - TODAY(*SAS-date*)

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

   - YEAR(*SAS-date*)
   - MONTH(*SAS-date*)
   - WEEKDAY(*SAS-date*)
   - TODAY(*SAS-date*)

3. 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 Country;
   Compensation=sum(Salary,Bonus);
run;
```

- 6
- 7
- 10
- None. This program contains a logic error.

3. 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 Country;
   Compensation=sum(Salary,Bonus);
run;
```

- 6
- 7
- 10
- None. This program contains a logic error.

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

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

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

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

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

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

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;
```

- 5
- 8
- 10
- It depends on the first value of **Type** in **orion.records**.

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;
```

- 5
- 8
- 10
- It depends on the first value of **Type** in **orion.records**.

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

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

PDV

Emp_ID	Salary	Ins	Bonus	Health_Award	Benefit
KBA	54000	800	250	.	.

- a missing value
- 55050
- 800
- 0

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

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

PDV

Emp_ID	Salary	Ins	Bonus	Health_Award	Benefit
KBA	54000	800	250	.	.

- a missing value
- 55050
- 800
- 0

7. Which of these statements does ***not*** correctly specify a SAS function?

   - Deadline=sum(TimeSpent, Last_Name);
   - GreatDay=today();
   - FingersToes=sum(10,10);
   - BirthdayYear=year('12dec1987'd);

7. Which of these statements does ***not*** correctly specify a SAS function?

   - Deadline=sum(TimeSpent, Last_Name);
   - GreatDay=today();
   - FingersToes=sum(10,10);
   - BirthdayYear=year('12dec1987'd);

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

- Yes
- No

```
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;
```

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

- (•) Yes
- ( ) No

```
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;
```

9. Which of the following determines the length of a new variable at compile time?

   a. INPUT statement

   b. assignment statement

   c. LENGTH statement

   d. all of the above

9. Which of the following determines the length of a new variable at compile time?

   a. INPUT statement

   b. assignment statement

   c. LENGTH statement

   d. all of the above

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

   - True
   - False

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

- True
- False