

\*Font that is in italics is what was changed in the text, unless otherwise specified.

## Content Corrections

Page Number	Item	Comment
3	Chapter 1, Instructions	The practice data for the SAS® Certified Specialist Prep Guide has been added to SAS OnDemand for Academics. See <a href="#">Instructions for Creating Prep Guide Data for SAS OnDemand for Academics</a> for the instructions to access the data.
56	Chapter 4 Quiz, Question 3	Correct Answer: B
78	Chapter 6, Removing the OBS Column	The data set used in the example code should be cert.admit and not work.example.  Updated complete code: <pre>proc print data=cert.admit noobs;   var age height weight fee; run;</pre>
85	Chapter 6, Example: FIRSTOBS= and OBS= as Data Set Options	The data set used in the example code should be cert.heart and not clinic.heart.  Updated complete code: <pre>options firstobs=10 obs=15; proc print data=cert.heart; run;</pre>
86	Chapter 6, Example: FIRSTOBS= and OBS= as Data Set Options	The data set should be cert.heart, and the values of the data set options should be firstobs=10 and obs=15.  Updated complete code: <pre>options firstobs=10 obs=15; proc print data=cert.heart(firstobs=10 obs=15); run;</pre>
117	Chapter 7, Iterations of the DATA Step	The section, “Iterations of the DATA Step,” has been updated to clarify and to correct images. See <a href="#">Corrected Section: Iterations of the DATA Step</a> later in this document.
145	Chapter 9, Date Constants	Updated syntax: <i>‘ dmmmyy’ d</i> or <i>‘ dmmmyyyy’ d</i>
181	Chapter 10, Example: Using One-to-One Reading to Combine Data Sets	Update to the last paragraph:  The resulting data set, Work.One2one, contains <i>nine</i> observations (the number of observations read from the smallest data set, which is Cert.Measure).
197	Chapter 10, Selecting Matching Observations	Update to the last paragraph:  In previous examples, Work.Merged contained 12 observations. In the output below, notice that only <i>11</i> observations met the condition in the IF expression.

\*Font that is in italics is what was changed in the text, unless otherwise specified.

278	Chapter 14, Example: Create New Name Variables	Updated complete DATA step:  <pre>data work.newnames(drop=name;   set cert.staff;   LastName=scan(name,1);   MiddleName=scan(name,3);   FirstName=scan(name,2); run;</pre>
281	Chapter 14, Example: SUBSTR Function	Updated complete code:  <pre>data work.agencyemp(drop=middlename);   set cert.agencyemp;   MiddleInitial=substr(middlename,1,1); run; proc print data=work.agencyemp; run;</pre>
300	Chapter 14, CEIL and FLOOR Functions	The lead-in sentence for the bulleted list should read:  Use these functions to round decimal arguments to the nearest integer:
366	Chapter 17, Scenario 2	Directions—updated second bullet point: <ul style="list-style-type: none"> <li>Remove observations with RestHR values that are greater than 70.</li> </ul>
379	Appendix 1, Chapter 4, Question 3	Correct Answer: B  Updated description: Use the OBS= option in the OPTIONS statement before the IMPORT procedure to limit the number of observations that SAS reads from the external file. When you use the OBS= option in the PROC PRINT statement, the whole file is imported but printing is limited to the number of observations specified. Use the DELIMITER='.' to indicate that the delimiter is a period(.) and use GETNAMES=YES to read in the first line, which contains the variable names.
381	Appendix 1, Chapter 5, Question 6	Updated description:  There is an equal sign between the DATA statement and the data set work.mysales. The equal sign should not be included when creating a new SAS data set and will generate a syntax error. The correct syntax for the DATA statement shown in B is:  <pre>data work.mysales;</pre>
385	Appendix 1, Chapter 9, Question 3	Updated description: You must enclose character values in quotation marks, and you must specify them in the same case in which they appear in the data set. The value of OK for Status in the IF-THEN statement matches the value in the data set, so the value of Count is incremented from 12 to 13. In the second IF-THEN statement, the value of E for Action matches the value in the data set, so the value of CONTROL is changed from Go to Stop.
386	Appendix 1, Chapter 10, Question 6	Updated description:  So, answers <i>b</i> and <i>c</i> , which contain unmatched observations, are incorrect.

\*Font that is in italics is what was changed in the text, unless otherwise specified.

395	Appendix 2, Scenario 2	The #6 comment for the PROC PRINT statement is missing an ending slash. The code should be:  <pre>proc print data=work.stress1; /* #6 */</pre>
395	Appendix 2, Scenario 2	Updated complete code:  <pre>data work.stress1; /*#1*/   set cert.stress; /*#2*/   where RestHR &lt; 70; /*#3*/   TotalTime=(timemin*60)+timesec; /*#4*/   if TotalTime&lt;600 then delete; /*#5*/ run; proc print data=work.stress1; /*#6*/ run;</pre> Updated code description for #3:  The WHERE statement selects only the observations where the values of RestHR are less than or equal to 70.
419	Index	VALIDNARNAME should be VALIDVARNAME.

## Enhancements Based on Feedback

Page Number	Item	Comment
126	Chapter 7 Quiz, Question 1	Rewording of the question:  Which of the following is not <i>created</i> during the compilation phase?
128	Chapter 7 Quiz, Question 10	Rewording of the question:  Which procedure produces distinct values of variables and can be used to <i>help</i> clean your data?
130	Chapter 8, Determine Whether the Data Requires Preprocessing	Added the following sentence:  Verify that your data is sorted on the BY variable that you want to use for BY-group processing.
137	Chapter 8 Quiz, Question 4	A PROC PRINT step was added to options b, c, and d. Without the PROC PRINT step, the data would not be displayed in HTML output. However, you can view your sorted data set in your Cert library.

## Corrections to the Practice Data

Data Set or Data File Name	SAS File that Creates the Data Set or Data File	Comment
stock.xlsx	cre8flatfile.sas	Added code to create stock.xlsx on 08Nov2022
trials	cre8permdata.sas	Added code to create trials data set on 08Nov2022

\*Font that is in italics is what was changed in the text, unless otherwise specified.

## Instructions for Creating Prep Guide Data for SAS OnDemand for Academics

These instructions are for the following certification guides:

- *SAS® Certified Specialist Prep Guide Base Programming Using SAS® 9.4*
- *SAS® Certification Prep Guide: Base Programming for SAS® 9, Fifth Edition*

1. Navigate to <https://support.sas.com/content/dam/SAS/support/en/books/data/base-guide-practice-data.zip> , download the ZIP file to a location on your computer such as your Downloads folder.
2. Unzip the file and save the unzipped files to a location on your computer so you can find the location again.
3. Log on to SAS OnDemand for Academics and open SAS Studio. Click on the **Server File and Folders** pane to make new folders.
4. In your SAS Studio session, right-click **Files (Home)** and select **New > Folder**.
  - a. Name your folder **base-guide-practice-data**.
  - b. Click **Save**.
5. Right-click the new **base-guide-practice-data** folder and select **New > Folder**.
  - a. Name the folder **cert**.
  - b. Click **Save**.
6. Right-click the new **cert** folder and select **Upload Files ...** .
  - a. Click **Choose Files**.
  - b. Navigate to the **base-guide-practice-data/cert** folder on your local machine where you saved the unzipped contents.
  - c. Select all files (Ctrl-A) and click **Open**. All files will appear in the Upload Files window.
  - d. Click **Upload**.
7. Right-click the new **base-guide-practice-data** folder and select **New > Folder**.
  - a. Name the folder **ehs**.
  - b. Click **Save**.
8. Right-click the **ehs** folder and select **Upload Files ...** .
  - a. Click **Choose Files**.
  - b. Navigate to the **base-guide-practice-data/ehs** folder on your local machine.
  - c. Select all files (Ctrl-A) and click **Open**. All files will appear in the Upload Files window.
  - d. Click **Upload**.
9. Right-click the new **base-guide-practice-data** folder and select **Upload Files ...** .
  - a. Click **Choose Files**.
  - b. Navigate to the **base-guide-practice-data** folder on your local machine.
  - c. Select **readme.txt**.
  - d. Click **Upload**.
10. In SAS OnDemand for Academics, click the **cert** folder.

\*Font that is in italics is what was changed in the text, unless otherwise specified.

## Corrected Section: Iterations of the DATA Step

### Iterations of the DATA Step

You can see that the DATA step works like a loop, repetitively executing statements to read data values and create observations one by one. At the beginning of the second iteration, the value of `_N_` is 2, and `_ERROR_` is still 0. Each loop (or cycle of execution) is called an *iteration*.

**Figure 7.11 Iterations of the DATA Step**

```

data work.update;
  set cert.invent;
  Total=instock+backord;
  SalePrice=(CostPerUnit*0.65)+CostPerUnit;
  format CostPerUnit SalePrice dollar6.2;
run;
    
```

As the SET statement executes for the second time, the values from the second record in CERT.INVENT are read from the input table into the PDV. The WORK.UPDATE output data set holds only the new observation calculated in the first iteration of the DATA step program. Notice that the PDV now holds the CERT.INVENT row 2 values for the Item, IDnum, InStock, BackOrd, and CostPerUnit variables. The values for Total and SalePrice have been initialized to missing.

**Figure 7.12 Program Data Vector and Output Data Set**

Program Data Vector								
Item	IDnum	InStock	BackOrd	CostPerUnit	Total	SalePrice	_N_	_ERROR_
6 Glass Mugs	SB082	6	12	\$1.50	.	.	2	0

SAS Data Set Work.Update Output							
Obs	Item	IDnum	InStock	BackOrd	CostPerUnit	Total	SalePrice
1	Bird Feeder	LG088	3	20	\$5.00	23	\$8.25



\*Font that is in italics is what was changed in the text, unless otherwise specified.

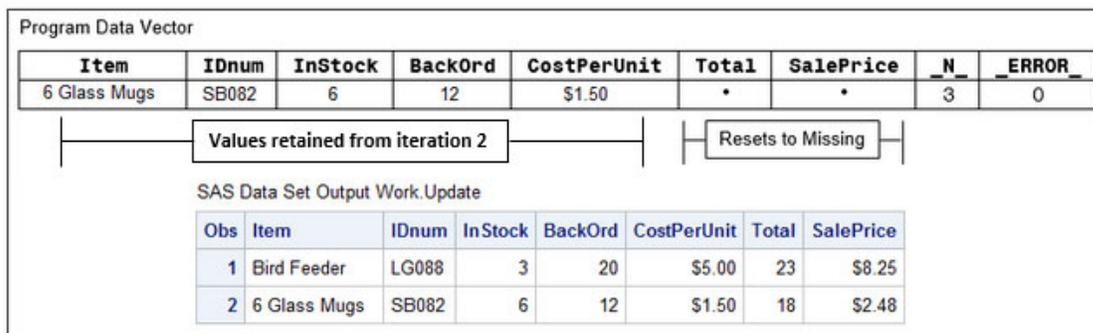
The RUN statement indicates the end of the DATA step loop. At the bottom of the DATA step, the values in the PDV are written to the data set as the second observation. The WORK.UPDATE data set has 2 observations after the RUN statement, as shown in Figure 7.14.

```
data work.update;
  set cert.invent;
  Total=instock+backord;
  SalePrice=(CostPerUnit*0.65)+CostPerUnit;
  format CostPerUnit SalePrice dollar6.2;
run;
```

Next, the value of `_N_` is incremented from 2 to 3. Control returns to the top of the DATA step. The values for `Item`, `IDnum`, `InStock`, `BackOrd`, and `CostPerUnit` are retained. The values for `Total` and `SalePrice` are reset to missing. The SET statement has an implied RETAIN statement within it to retain the original values of the data set.

```
data work.update;
  set cert.invent;
  Total=instock+backord;
  SalePrice=(CostPerUnit*0.65)+CostPerUnit;
  format CostPerUnit SalePrice dollar6.2;
run;
```

**Figure 7.15 Program Data Vector and Output Data**



When PROC IMPORT reads raw data, SAS sets the value of each variable in the DATA step to missing at the beginning of each cycle of execution, with these exceptions:

- variables that are named in a RETAIN statement
- variables that are created in a sum statement
- automatic variables

In contrast, when reading variables from a SAS data set, SAS sets the values to missing only before the first cycle of execution of the DATA step. Therefore, the variables retain their values until new values become available, as shown in Figure 7.15. In Figure 7.15, when `_N_` increases from 2 to 3 at the top of the DATA step, only `Total` and `SalePrice` are set to missing. The other variables, which come from the CERT.INVENT data set, will only change through an assignment statement or through the next execution of a SET or MERGE statement. Variables that are created with options in a SET or MERGE statement also retain their values from one cycle of execution to the next.