

SAS ADVANCED PROGRAMMING CERTIFICATION PRACTICE EXAM (SAMPLE) STUDY GUIDE



SAMPLE STUDY GUIDE WITH LIMITED QUESTIONS

VISIT US ONLINE FOR
THE FULL VERSION STUDY GUIDE

<https://sasadvprogramming.examzify.com>



Copyright © 2026 by Examzify - A Kaluba Technologies Inc. product.

ALL RIGHTS RESERVED.

No part of this book may be reproduced or transferred in any form or by any means, graphic, electronic, or mechanical, including photocopying, recording, web distribution, taping, or by any information storage retrieval system, without the written permission of the author.

Notice: Examzify makes every reasonable effort to obtain accurate, complete, and timely information about this product from reliable sources.

SAMPLE

Table of Contents

Copyright	1
Table of Contents	2
Introduction	3
How to Use This Guide	4
Questions	5
Answers	8
Explanations	10
Next Steps	16

SAMPLE

Introduction

Preparing for a certification exam can feel overwhelming, but with the right tools, it becomes an opportunity to build confidence, sharpen your skills, and move one step closer to your goals. At Examzify, we believe that effective exam preparation isn't just about memorization, it's about understanding the material, identifying knowledge gaps, and building the test-taking strategies that lead to success.

This guide was designed to help you do exactly that.

Whether you're preparing for a licensing exam, professional certification, or entry-level qualification, this book offers structured practice to reinforce key concepts. You'll find a wide range of multiple-choice questions, each followed by clear explanations to help you understand not just the right answer, but why it's correct.

The content in this guide is based on real-world exam objectives and aligned with the types of questions and topics commonly found on official tests. It's ideal for learners who want to:

- Practice answering questions under realistic conditions,
- Improve accuracy and speed,
- Review explanations to strengthen weak areas, and
- Approach the exam with greater confidence.

We recommend using this book not as a stand-alone study tool, but alongside other resources like flashcards, textbooks, or hands-on training. For best results, we recommend working through each question, reflecting on the explanation provided, and revisiting the topics that challenge you most.

Remember: successful test preparation isn't about getting every question right the first time, it's about learning from your mistakes and improving over time. Stay focused, trust the process, and know that every page you turn brings you closer to success.

Let's begin.

How to Use This Guide

This guide is designed to help you study more effectively and approach your exam with confidence. Whether you're reviewing for the first time or doing a final refresh, here's how to get the most out of your Examzify study guide:

1. Start with a Diagnostic Review

Skim through the questions to get a sense of what you know and what you need to focus on. Your goal is to identify knowledge gaps early.

2. Study in Short, Focused Sessions

Break your study time into manageable blocks (e.g. 30 – 45 minutes). Review a handful of questions, reflect on the explanations.

3. Learn from the Explanations

After answering a question, always read the explanation, even if you got it right. It reinforces key points, corrects misunderstandings, and teaches subtle distinctions between similar answers.

4. Track Your Progress

Use bookmarks or notes (if reading digitally) to mark difficult questions. Revisit these regularly and track improvements over time.

5. Simulate the Real Exam

Once you're comfortable, try taking a full set of questions without pausing. Set a timer and simulate test-day conditions to build confidence and time management skills.

6. Repeat and Review

Don't just study once, repetition builds retention. Re-attempt questions after a few days and revisit explanations to reinforce learning. Pair this guide with other Examzify tools like flashcards, and digital practice tests to strengthen your preparation across formats.

There's no single right way to study, but consistent, thoughtful effort always wins. Use this guide flexibly, adapt the tips above to fit your pace and learning style. You've got this!

Questions

SAMPLE

1. Which statement is false concerning the following array statement: `array TwoD[2,3] $ _temporary_;`
 - A. Six array elements are created in the PDV
 - B. The array elements are automatically retained
 - C. The array elements will not be in the final output table
 - D. The array elements can be referenced using one number
2. In what circumstance would you use the NODUPKEY option?
 - A. When generating graphs
 - B. When wanting to keep duplicate records
 - C. When removing duplicates based on key variables
 - D. When merging datasets
3. What is the primary purpose of a temporary array in SAS?
 - A. To store values that will be used in calculations only within a data step
 - B. To create columns that will be saved in the final dataset
 - C. To automatically output dataset rows
 - D. To index a dataset for efficient data retrieval
4. Which of the following is not a valid way to specify the output column label in PROC SQL?
 - A. `Label='Employee ID'`
 - B. `'Employee Number'`
 - C. `Label= 'Employee Number'`
 - D. `Header= 'Employee ID'`
5. What is the purpose of the SAS Output Delivery System (ODS)?
 - A. To execute SAS programs in batch mode
 - B. To optimize memory usage during processing
 - C. To manage and customize output formats
 - D. To create user-defined formats for variables

6. What type of data will the LENGTH statement affect?

- A. Text data only
- B. Numerical data only
- C. Both numerical and character data
- D. Data structures only

7. Which statement is used to read a CSV file in SAS?

- A. DATA statement
- B. INFILE statement with the DSD option
- C. IMPORT statement
- D. FILENAME statement

8. What is the purpose of the RANK procedure in SAS?

- A. To sort data in ascending order
- B. To assign ranks to data values within a variable
- C. To remove duplicates from a dataset
- D. To calculate the mean of a variable

9. Which SAS procedure is specifically designed for generating reports from datasets?

- A. PROC PRINT
- B. PROC SORT
- C. PROC FREQ
- D. PROC CONTENTS

10. What is the use of the WHERE statement in PROC SQL?

- A. To define new variables based on conditions
- B. To filter rows based on specific conditions
- C. To format the output results of a query
- D. To create indexes for a dataset

Answers

SAMPLE

1. D
2. C
3. A
4. D
5. C
6. C
7. B
8. B
9. A
10. B

SAMPLE

Explanations

SAMPLE

1. Which statement is false concerning the following array statement: array TwoD[2,3] \$ _temporary_?

- A. Six array elements are created in the PDV
- B. The array elements are automatically retained
- C. The array elements will not be in the final output table
- D. The array elements can be referenced using one number**

The statement regarding the array declaration, "array TwoD[2,3] \$ _temporary_," creates a temporary two-dimensional array with 2 rows and 3 columns, meaning a total of 6 elements are indeed created. This determines that six separate array elements exist in the program data vector (PDV). When it comes to the nature of temporary arrays in SAS, these elements are automatically retained between iterations within the DATA step but will not be included in any output data set, which corresponds to the properties of temporary arrays. Because the array is temporary, once the DATA step is completed, its elements vanish and are not part of the final output table created. In this context, the way to access elements in a two-dimensional array differs from a one-dimensional array. For the declared array, you need to reference elements using two indices because it has both row and column dimensions. Therefore, suggesting that the array elements can be referenced with a single number is inaccurate since a two-dimensional array requires accessing it in a row-column format (e.g., TwoD[1,2] to access the element in the first row and second column). This is why the statement about being able to reference one number is false.

2. In what circumstance would you use the NODUPKEY option?

- A. When generating graphs
- B. When wanting to keep duplicate records
- C. When removing duplicates based on key variables**
- D. When merging datasets

The NODUPKEY option is specifically designed to address situations where you want to eliminate duplicate records based on key variables in a dataset. When this option is used, SAS examines the designated key variables and keeps only the first occurrence of each unique combination, essentially filtering out any subsequent duplicate rows. This is particularly useful in data processing tasks where maintaining distinct records is essential for analysis, such as when preparing data for reporting or statistical modeling. In scenarios involving graphs, the need to remove duplicates based on key variables does not apply directly, as the focus there is on visual representation rather than on data integrity regarding duplicates. Additionally, wanting to keep duplicate records contradicts the purpose of using NODUPKEY, as its primary function is to purge duplicates. In the context of merging datasets, while handling duplicates may become relevant, the NODUPKEY option is not specifically tied to the merging process itself but rather to the unique identification of records within a dataset. Thus, the circumstance in which NODUPKEY is most appropriately applied is indeed when removing duplicates based on key variables.

3. What is the primary purpose of a temporary array in SAS?

- A. To store values that will be used in calculations only within a data step
- B. To create columns that will be saved in the final dataset
- C. To automatically output dataset rows
- D. To index a dataset for efficient data retrieval

The primary purpose of a temporary array in SAS is indeed to store values that will be used in calculations only within a data step. Temporary arrays are created for use within a single data step and do not retain their values or structure once the data step is completed. This characteristic makes them particularly useful for intermediate calculations, transformations, or to hold values that are needed only during the processing of that specific data step. When working with temporary arrays, you can manipulate data efficiently by keeping it in memory without creating additional permanent datasets or columns. This allows for faster computations and cleaner code, as the data manipulated in a temporary array does not clutter the final output dataset. As soon as the data step ends, the array and its content are cleared from memory. In contrast, other options may suggest functionality not associated with temporary arrays. For example, creating columns saved in the final dataset is typically the role of permanent arrays or direct variable assignments in the data step. Automatic output of dataset rows is managed by the data step itself, while indexing for efficient data retrieval is a separate operation applied at the dataset level rather than within the context of a temporary array. Thus, the correct understanding is that temporary arrays are specifically for temporary storage and calculation within a single data step context.

4. Which of the following is not a valid way to specify the output column label in PROC SQL?

- A. Label='Employee ID'
- B. 'Employee Number'
- C. Label= 'Employee Number'
- D. Header= 'Employee ID'

In PROC SQL, specifying output column labels can be done using the LABEL option or directly with quoted strings. The option that is not valid for specifying an output column label is the approach that uses the HEADER keyword, as there is no HEADER option for labeling columns in PROC SQL. The LABEL statement specifically allows you to assign descriptive labels to columns that enhance readability in output. This can be done by assigning a label to a variable name or directly within a SELECT statement, making options that utilize the LABEL keyword or quoted strings valid methods. For instance, using Label='Employee ID' or Label='Employee Number' correctly employs the LABEL option to set the intended column label. Similarly, quoting a label such as 'Employee Number' is also valid, as PROC SQL recognizes quoted strings as appropriate labels for output. In contrast, the HEADER option lacks validity as it is not recognized in the context of column labeling within PROC SQL, thus making it the incorrect choice among the given options.

5. What is the purpose of the SAS Output Delivery System (ODS)?

- A. To execute SAS programs in batch mode
- B. To optimize memory usage during processing
- C. To manage and customize output formats**
- D. To create user-defined formats for variables

The SAS Output Delivery System (ODS) serves a crucial role in managing and customizing the output generated by SAS procedures. Its primary function is to allow users to produce output in various formats, such as HTML, PDF, Excel, and RTF, among others. By leveraging ODS, users can apply styles, select specific output elements, and direct outputs to different destinations. This capability makes it possible to create highly customized reports that meet specific requirements or presentation standards. In contrast, executing programs in batch mode focuses more on how SAS programs are run rather than how output is formatted. Optimizing memory usage is a separate consideration related to performance during data processing, while user-defined formats pertain to data representation rather than to the output formatting and delivery system. Therefore, the ability of ODS to manage and customize output formats distinctly defines its purpose.

6. What type of data will the LENGTH statement affect?

- A. Text data only
- B. Numerical data only
- C. Both numerical and character data**
- D. Data structures only

The LENGTH statement in SAS is designed to specify the length of variables, and it can be applied to both character and numerical data types. For character variables, the LENGTH statement determines the maximum number of bytes that can be used to store that variable. For instance, if you set the length of a character variable to 20, it can hold up to 20 characters. For numerical variables, the LENGTH statement can also have an effect, although it is less common to specify length for numeric variables. In SAS, the storage of numeric types (like integers or floating-point numbers) is typically based on the type of numeric data and the amount of memory allocated by SAS for those numbers, which can adapt dynamically. However, the LENGTH statement can still influence aspects like the precision of the numeric data, particularly when utilizing formats to control how numeric data is displayed. Thus, when considering the implications of the LENGTH statement, it can affect both character and numerical data, making the correct choice encompass both types.

7. Which statement is used to read a CSV file in SAS?

- A. DATA statement
- B. INFILE statement with the DSD option**
- C. IMPORT statement
- D. FILENAME statement

The INFILE statement with the DSD option is the appropriate choice for reading a CSV file in SAS because it allows for the effective handling of delimited data formats, specifically CSV files, which utilize commas as the delimiter. The DSD option is particularly useful as it ensures that consecutive delimiters are treated as missing values and allows for the proper handling of quoted strings, which could contain delimiters within the text. This is essential when reading CSV files, since improper handling of these cases could lead to incorrect data interpretation. The DATA statement is generally used to create or manipulate datasets within SAS but does not directly facilitate the reading of CSV files on its own. While the IMPORT statement can be employed to read a CSV file as well, it is more often used for general-purpose import tasks and may not offer the same level of detail or customization as utilizing the INFILE statement with options like DSD specifically tailored for CSV formats. The FILENAME statement serves to assign a logical name to a file, but does not read the data directly; instead, it is used primarily in conjunction with other statements to provide a reference to a specific file location.

8. What is the purpose of the RANK procedure in SAS?

- A. To sort data in ascending order
- B. To assign ranks to data values within a variable**
- C. To remove duplicates from a dataset
- D. To calculate the mean of a variable

The RANK procedure in SAS is specifically designed to assign ranks to data values within one or more variables. This procedure is particularly useful in statistical analysis and data processing when you want to understand the relative standing of values within a given dataset. For instance, if you have a dataset of test scores and you want to determine which scores are the highest, the RANK procedure would take care of that by providing a ranking for each test score, such that the highest score receives the top rank. The RANK procedure can accommodate various ranking methods, including handling ties, where it can assign the same rank to equal values or use average ranks as needed. It allows users to create ranked lists that can facilitate further analysis or reporting tasks. The other options address different functionalities that are not related to the primary purpose of the RANK procedure. Sorting data in ascending order pertains to the SORT procedure, removing duplicates relates to the SORT procedure with the option to remove duplicates, and calculating the mean of a variable is handled by procedures like MEANS or SUMMARY, rather than the RANK procedure.

9. Which SAS procedure is specifically designed for generating reports from datasets?

- A. PROC PRINT**
- B. PROC SORT
- C. PROC FREQ
- D. PROC CONTENTS

The procedure designed specifically for generating reports from datasets is PROC PRINT. This procedure is used to display the contents of a SAS dataset in a tabular format, making it ideal for producing straightforward reports that allow users to view data as it is structured in the dataset. By default, PROC PRINT outputs all observations and variables, but it also provides options to customize the display, such as selecting specific variables or formatting the output. In contrast, the other procedures serve different purposes. PROC SORT is primarily used for ordering a dataset based on specified variables; it is essential for organizing data but does not generate reports. PROC FREQ analyzes categorical data and provides frequency counts, which can be useful for summary reports but is not intended for general dataset reporting. PROC CONTENTS gives metadata about a dataset, including information on variables and their attributes, but it does not present actual data records in a report format. Thus, PROC PRINT stands out as the dedicated procedure for generating data reports in SAS.

10. What is the use of the WHERE statement in PROC SQL?

- A. To define new variables based on conditions
- B. To filter rows based on specific conditions**
- C. To format the output results of a query
- D. To create indexes for a dataset

The WHERE statement in PROC SQL is primarily utilized to filter rows based on specific conditions. This functionality allows the user to specify criteria that determine which records are included in the result set of the SQL query. By using the WHERE clause, you can narrow down the data to only those rows that meet your specified conditions. For example, if you're interested in retrieving records of employees with a salary greater than a specific amount, the WHERE clause would enable you to extract only those relevant entries from the dataset. In contrast, the other provided options serve different purposes. Defining new variables based on conditions is typically handled through the CASE statement or other data manipulation techniques, rather than the WHERE statement. Formatting output results is generally achieved using the FORMAT statement or attributes in SQL, rather than filtering directly. Creating indexes for a dataset involves methods that are focused on improving query performance but is not related to the WHERE statement's functionality.

SAMPLE

Next Steps

Congratulations on reaching the final section of this guide. You've taken a meaningful step toward passing your certification exam and advancing your career.

As you continue preparing, remember that consistent practice, review, and self-reflection are key to success. Make time to revisit difficult topics, simulate exam conditions, and track your progress along the way.

If you need help, have suggestions, or want to share feedback, we'd love to hear from you. Reach out to our team at hello@examzify.com.

Or visit your dedicated course page for more study tools and resources:

<https://sasadvprogramming.examzify.com>

We wish you the very best on your exam journey. You've got this!

SAMPLE