



BrainDumps
Collection

SAP

C_ABAPD_2309 Exam

SAP Certified Associate - Back-End Developer - ABAP Cloud

Thank you for Downloading C_ABAPD_2309 exam PDF Demo

You can also try our C_ABAPD_2309 practice exam software

Download Free Demo

https://www.braindumpscollection.com/C_ABAPD_2309.html

DEMO
VERSION

(LIMITED CONTENT)

**Questions
& Answers**

Version: 4.2

Question: 1

Which function call returns 0?

- A. Count_any_of (val - 'ABAP ABAP abap' sub "AB")
- B. Count (val - 'ABAP ABAP abap' sub - 'AB')
- C. find_any_of (val = "ABAP ABAP abap' sub = "AB")
- D. find_any_not_of(val 'ABAP ABAP abap' sub = 'AB')

Answer: D

Explanation:

The function find_any_not_of returns the position of the first character in the string val that is not contained in the string sub. If no such character is found, the function returns 0. In this case, the string val contains only the characters A, B, and a, which are all contained in the string sub, so the function returns 0. The other functions return positive values, as follows:

Count_any_of returns the number of occurrences of any character in the string sub within the string val. In this case, it returns 8, since there are 8 A's and B's in val.

Count returns the number of occurrences of the string sub within the string val. In this case, it returns 2, since there are 2 AB's in val.

find_any_of returns the position of the first character in the string val that is contained in the string sub. In this case, it returns 1, since the first character A is in sub. Reference: [String Functions - ABAP Keyword Documentation](#), [Examples of String Functions - ABAP Keyword Documentation](#)

Question: 2

In which products must you use the ABAP Cloud Development Model? Note: There are 2 correct answers to this question.

- A. SAP S/4HANA Cloud, private edition
- B. SAP BTP, ABAP environment
- C. SAP S/4HANA on premise
- D. SAP S/4HANA Cloud, public edition

Answer: AB

Explanation:

The ABAP Cloud Development Model is the ABAP development model to build cloud-ready business apps, services, and extensions. It comes with SAP BTP and SAP S/4HANA. [It works with public or](#)

private cloud, and even on-premise¹. However, the complete ABAP Cloud Development Model, including the cloud-optimized ABAP language and public local SAP APIs and extension points, is available only in SAP BTP ABAP Environment and in the 2208/2022 versions of the SAP S/4HANA editions¹. Therefore, you must use the ABAP Cloud Development Model in SAP BTP, ABAP environment and SAP S/4HANA Cloud, private edition. You can also use it in SAP S/4HANA on premise, but it is not mandatory. You cannot use it in SAP S/4HANA Cloud, public edition, because it does not allow custom ABAP code². Reference: 1: [ABAP Cloud](#) | [SAP Blogs](#) 2: [SAP S/4HANA Cloud Extensibility – Overview and Comparison](#) | [SAP Blogs](#)

Question: 3

Which RESTful Application Programming object can be used to organize the display of fields in an app?

- A. Data model view
- B. Metadata extension
- C. Service definition
- D. Projection view

Answer: B

Explanation:

A metadata extension is a RESTful Application Programming object that can be used to organize the display of fields in an app. A metadata extension is a CDS view that annotates another CDS view with UI annotations, such as labels, icons, or facets. These annotations define how the data should be presented in the app, such as which fields should be shown on the object page, which fields should be editable, or which fields should be used for filtering or sorting. [A metadata extension can also be used to add custom actions or validations to the app](#)¹². Reference: 1: [Refine the Object Page with Annotations](#) | [SAP Tutorials](#) 2: [ABAP RAP : Enabling custom actions with a dialog for additional input fields](#) | [SAP Blogs](#)

Question: 4

Refer to the Exhibit.

```
DATA: go_super TYPE REF TO lcl_super,  
      go_sub   TYPE REF TO lcl_sub.  
go_sub = NEW #( ... ).  
go_super = go_sub.  
  
with lcl_super being superclass of lcl_sub.
```

When accessing the subclass instance through go_super, what can you do? Note: There are 2 correct answers to this question.

- A. Access the inherited private components.
- B. Access the inherited public components.
- C. Call a subclass specific public method
- D. Call inherited public redefined methods.

Answer: AB

Explanation:

When accessing the subclass instance through `go_super`, you can do both of the following:

Access the inherited private components: A subclass inherits all the private attributes and methods of its superclass, unless they are explicitly overridden by the subclass. Therefore, you can access the inherited private components of the superclass through `go_super`, as long as they are not hidden by other attributes or methods in the subclass12.

Access the inherited public components: A subclass inherits all the public attributes and methods of its superclass, unless they are explicitly overridden by the subclass. Therefore, you can access the inherited public components of the superclass through `go_super`, as long as they are not hidden by other attributes or methods in the subclass12.

You cannot do any of the following:

Call a subclass specific public method: A subclass does not have any public methods that are not inherited from its superclass. Therefore, you cannot call a subclass specific public method through `go_super12.`

Call inherited public redefined methods: A subclass does not have any public methods that are redefined from its superclass. Therefore, you cannot call inherited public redefined methods through `go_super12.`

Reference: 1: [Object Oriented - ABAP Development - Support Wiki](#) 2: [Inheritance and Instantiation - ABAP Keyword Documentation](#)

Question: 5

Given the following code in an SAP S/4HANA Cloud private edition tenant:

```
1 CLASS zcl_demo_class DEFINITION.  
2 METHODS: m1.  
3 ENDCLASS..  
4 CLASS zcl_demo_class_Implementation.  
5 METHOD m1.  
6 CALL FUNCTION 'ZF1'.  
7 ENDMETHOD  
8 ENDCLASS.
```

The class `zcl_demo_class` is in a software component with the language version set to "ABAP Cloud". The function module `ZF1` is in a different software component with the language version set to "Standard ABAP". Both the class and function module are customer created.

Regarding line #6, which of the following are valid statements? Note: There are 2 correct answers to this question.

- A. ZF1' can be called only if it is released for cloud development.
- B. 'ZF1' can be called if a wrapper is created for it and the wrapper itself is released for cloud development.
- C. "ZF1" can be called whether it is released or not for cloud development
- D. ZF1" can be called if a wrapper is created for it but the wrapper itself is not released for cloud development.

Answer: AB

Explanation:

The ABAP Cloud Development Model requires that only public SAP APIs and extension points are used to access SAP functionality and data. These APIs and extension points are released by SAP and documented in the SAP API Business Hub1. Customer-created function modules are not part of the public SAP APIs and are not released for cloud development. Therefore, calling a function module directly from an ABAP Cloud class is not allowed and will result in a syntax error. However, there are two possible ways to call a function module indirectly from an ABAP Cloud class:

Create a wrapper class or interface for the function module and release it for cloud development. A wrapper is a class or interface that encapsulates the function module and exposes its functionality through public methods or attributes. The wrapper must be created in a software component with the language version set to "Standard ABAP" and must be marked as released for cloud development using the annotation `@EndUserText.label`. The wrapper can then be called from an ABAP Cloud class using the public methods or attributes2.

Use the ABAP Cloud Connector to call the function module as a remote function call (RFC) from an ABAP Cloud class. The ABAP Cloud Connector is a service that enables the secure and reliable communication between SAP BTP, ABAP environment and on-premise systems. The function module must be exposed as an RFC-enabled function module in the on-premise system and must be registered in the ABAP Cloud Connector. The ABAP Cloud class can then use the class `cl_rfc_destination_service` to get the destination name and the class `cl_abap_system` to create a proxy object for the function module. The proxy object can then be used to call the function module3.

Reference: 1: SAP API Business Hub 2: Creating an ABAP Cloud Project | SAP Help Portal 3: Calling Remote Function Modules | SAP Help Portal

Thank You for trying C_ABAPD_2309 PDF Demo

To try our C_ABAPD_2309 practice exam software visit link below

https://www.braindumpscollection.com/C_ABAPD_2309.html

Start Your C_ABAPD_2309 Preparation

Use Coupon "20OFF" for extra 20% discount on the purchase of Practice Test Software. Test your C_ABAPD_2309 preparation with actual exam questions.