



Free Questions for C_ABAPD_2309 by certscare

Shared by Newton on 09-08-2024

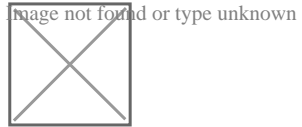
For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

Refer to the Exhibit.



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

Options:

- 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:

A, B

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_super`¹².

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_super`¹².

Question 2

Question Type: MultipleChoice

Refer to the Exhibit.

Image not found or type unknown



The class `zcl_demo_class` is in a software component with the language version set to 'Standard ABAP'. The function module 'ZF11' is in a software component with the language version set to 'ABAP Cloud'. Both the class and function module are customer created.

Regarding line #6, which of the following is a valid statement?

Options:

- A) 'ZF1' can be called whether it has been released or not for cloud development.
- B) 'ZF1' can be called via a wrapper that itself has been released for cloud development.
- C) 'ZF1' can be called via a wrapper that itself has not been released for cloud development.
- D) 'ZF1' must be released for cloud development to be called.

Answer:

B

Explanation:

The function module ZF1 is in a software component with the language version set to "ABAP Cloud". This means that it follows the ABAP Cloud Development Model, which requires the usage of public SAP APIs and extension points 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 a class with the language version set to "Standard ABAP" is not allowed and will result in a syntax error. However, there is a possible way to call a function module indirectly from a class with the language version set to "Standard ABAP":

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 "ABAP Cloud" and must be marked as released for cloud development using the annotation `@EndUserText.label`. The wrapper can then be called from a class with the language version set to "Standard ABAP" using the public methods or attributes².

For example, the following code snippet shows how to create a wrapper class for the function module ZF1 and call it from the class `zcl_demo_class`:

```
@EndUserText.label: 'Wrapper for ZF1' CLASS zcl_wrapper_zf1 DEFINITION PUBLIC FINAL CREATE PUBLIC. PUBLIC SECTION.  
CLASS-METHODS: call_zf1 IMPORTING iv_a TYPE i iv_b TYPE i EXPORTING ev_result TYPE i. ENDCLASS.
```

```
CLASS zcl_wrapper_zf1 IMPLEMENTATION. METHOD call_zf1. CALL FUNCTION 'ZF1' EXPORTING a = iv_a b = iv_b IMPORTING  
result = ev_result. ENDMETHOD. ENDCLASS.
```

```
CLASS zcl_demo_class DEFINITION. METHODS: m1. ENDCLASS.
```

```
CLASS zcl_demo_class IMPLEMENTATION. METHOD m1. DATA(lv_result) = zcl_wrapper_zf1=>call_zf1( iv_a = 2 iv_b = 3 ).  
WRITE: / lv_result. ENDMETHOD. ENDCLASS.
```

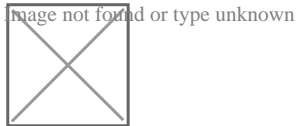
The output of this code is:

5

Question 3

Question Type: MultipleChoice

Refer to the Exhibit.



What are valid statements? Note: There are 2 correct answers to this question.

Options:

- A) 'zcxl' is a dictionary structure, and 'param1' and 'param2' are this structure.
- B) 'param11' and 'param2' are predefined names.

- C) The code creates an exception object and raises an exception.
- D) 'previous' expects the reference to a previous exception

Answer:

C, D

Explanation:

The code snippet in the image is an example of using the RAISE EXCEPTION statement to raise a class-based exception and create a corresponding exception object. The code snippet also uses the EXPORTING addition to pass parameters to the instance constructor of the exception class¹². Some of the valid statements about the code snippet are:

The code creates an exception object and raises an exception: This is true. The RAISE EXCEPTION statement raises the exception linked to the exception class zcxl and generates a corresponding exception object. The exception object contains the information about the exception, such as the message, the source position, and the previous exception¹².

"previous" expects the reference to a previous exception: This is true. The previous parameter is a predefined parameter of the instance constructor of the exception class cx_root, which is the root class of all class-based exceptions. The previous parameter expects the reference to a previous exception object that was caught during exception handling. The previous parameter can be used to chain multiple exceptions and preserve the original cause of the exception¹².

You cannot do any of the following:

"zcxl" is a dictionary structure, and "param1" and "param2" are this structure: This is false. zcxl is not a dictionary structure, but a user-defined exception class that inherits from the predefined exception class cx_static_check. param1 and param2 are not components of this structure, but input parameters of the instance constructor of the exception class zcxl. The input parameters can be used to pass additional information to the exception object, such as the values that caused the exception12.

"param1" and "param2" are predefined names: This is false. param1 and param2 are not predefined names, but user-defined names that can be chosen arbitrarily. However, they must match the names of the input parameters of the instance constructor of the exception class zcxl. The names of the input parameters can be declared in the interface of the exception class using the RAISING addition12.

Question 4

Question Type: MultipleChoice

Refer to the Exhibit.

Image not found or type unknown



Using ABAP SQL, which select statement selects the mat field on line #17?

Options:

- A) SELECT mat FROM Material...
- B) SELECT mat FROM demo_sales_cds_so_i_ve...
- C) SELECT mat FROM demo_sales_so_i...
- D) SELECT mat FROM demo sales cds material ve...

Answer:

B

Explanation:

Using ABAP SQL, the select statement that selects the mat field on line #17 is:

```
SELECT mat FROM demo_sales_cds_so_i_ve...
```

This statement selects the mat field from the CDS view demo_sales_cds_so_i_ve, which is defined on line #1. The CDS view demo_sales_cds_so_i_ve is a projection view that projects the fields of the CDS view demo_sales_cds_so_i, which is defined on line #2. The CDS view demo_sales_cds_so_i is a join view that joins the fields of the database table demo_sales_so_i, which is defined on line #3, and the CDS view demo_sales_cds_material_ve, which is defined on line #4. The CDS view demo_sales_cds_material_ve is a value help view that provides value help for the material field of the database table demo_sales_so_i. The mat field is an alias for the material field of the database table demo_sales_so_i, which is defined on line #91.

The other options are not valid because:

A) `SELECT mat FROM Material...` is not valid because `Material` is not a valid data source in the given code. There is no CDS view or database table named `Material`.

C) `SELECT mat FROM demo_sales_so_i...` is not valid because `demo_sales_so_i` is not a valid data source in the given code. There is no CDS view named `demo_sales_so_i`, only a database table. To access a database table, the keyword `TABLE` must be used, such as `SELECT mat FROM TABLE demo_sales_so_i...`

D) `SELECT mat FROM demo sales cds material ve...` is not valid because `demo sales cds material ve` is not a valid data source in the given code. There is no CDS view or database table named `demo sales cds material ve`. The correct name of the CDS view is `demo_sales_cds_material_ve`, with underscores instead of spaces.

Question 5

Question Type: MultipleChoice

Refer to the Exhibit.

Image not found or type unknown



Which of the following ON conditions must you insert in place of '????'?

Options:

- A) ON Z_Source1.carrier_id = Z_Source2.carrier_id
- B) ON Sprojection.carrier_id=Z_Source2.carrier_id
- C) ON Sprojection.carrier_id=Z_Source2.carrier
- D) ON Sprojection.carrier_id=Z_Source2.carrier_id

Answer:

D

Explanation:

The correct ON condition that must be inserted in place of "???" is:

ON Sprojection.carrier_id=Z_Source2.carrier_id

This ON condition specifies the join condition between the CDS view Sprojection and the database table Z_Source2. The join condition is based on the field carrier_id, which is the primary key of both the CDS view and the database table. The ON condition ensures that only the records that have the same value for the carrier_id field are joined together.

The other options are not valid ON conditions, because:

A) ON Z_Source1.camer_id = 7_Source2 carrier_id is not valid because Z_Source1 and 7_Source2 are not valid data sources in the given code. There is no CDS view or database table named Z_Source1 or 7_Source2. The correct names are Z_Source1 and Z_Source2. Moreover, the field camer_id is not a valid field in the given code. There is no field named camer_id in any of the data sources. The correct name is carrier_id.

B) ON Sprojection Camer=Source2 carrier_id is not valid because Sprojection and Source2 are not valid data sources in the given code. There is no CDS view or database table named Sprojection or Source2. The correct names are Sprojection and Z_Source2. Moreover, the field Camer is not a valid field in the given code. There is no field named Camer in any of the data sources. The correct name is carrier_id. Furthermore, the ON condition is missing the dot (.) operator between the data source name and the field name, which is required to access the fields of the data source1.

C) ON Sprojection. Carrier Source2.carrier is not valid because Carrier and carrier are not valid fields in the given code. There is no field named Carrier or carrier in any of the data sources. The correct name is carrier_id. Moreover, the ON condition is missing the dot (.) operator between the data source name and the field name, which is required to access the fields of the data source1.

Question 6

Question Type: MultipleChoice

What is the sequence priority when evaluating a logical expression?

A) NOT 1

B) OR 3

C) AND 2

Options:

A) A B C

B) CAB

C) A C B

D) B A C

Answer:

C

Explanation:

The sequence priority when evaluating a logical expression is C. A C B, which means NOT, AND, OR. This is the order of precedence of the Boolean operators in ABAP, which determines how the system implicitly parenthesizes all logical expressions that are not closed by explicit parentheses. The operator with the highest priority is evaluated first, and the operator with the lowest priority is evaluated last. The order of precedence of the Boolean operators in ABAP is as follows:

NOT: The NOT operator is a unary operator that negates the logical expression that follows it. It has the highest priority and is evaluated before any other operator. For example, in the expression NOT a AND b, the NOT operator is applied to a first, and then the AND operator is applied to the result and b.

AND: The AND operator is a binary operator that returns true if both logical expressions on its left and right are true, and false otherwise. It has the second highest priority and is evaluated before the OR and EQUIV operators. For example, in the expression a AND b OR c, the AND operator is applied to a and b first, and then the OR operator is applied to the result and c.

OR: The OR operator is a binary operator that returns true if either or both logical expressions on its left and right are true, and false otherwise. It has the third highest priority and is evaluated after the NOT and AND operators, but before the EQUIV operator. For example, in the expression a OR b EQUIV c, the OR operator is applied to a and b first, and then the EQUIV operator is applied to the result and c.

EQUIV: The EQUIV operator is a binary operator that returns true if both logical expressions on its left and right have the same truth value, and false otherwise. It has the lowest priority and is evaluated after all other operators. For example, in the expression a AND b EQUIV c OR d, the EQUIV operator is applied to a AND b and c last, after the AND and OR operators are applied.

Question 7

Question Type: MultipleChoice

Refer to the Exhibit.

Image not found or type unknown



What are valid statements? Note: There are 3 correct answers to this question

Options:

- A) In class CL1, the interface method is named if-ml.
- B) Class CL2 uses the interface.
- C) Class CL1 uses the interface.
- D) In class CL2, the interface method is named ifl-ml.
- E) Class CL1 implements the interface.

Answer:

B, D, E

Explanation:

The following are the explanations for each statement:

C: This statement is valid. Class CL1 uses the interface. This is because class CL1 implements the interface ifl using the INTERFACES statement in the public section of the class definition. The INTERFACES statement makes the class compatible with the interface and inherits all the components of the interface. The class can then use the interface components, such as the method ml, by using the interface component selector ~, such as ifl~ml¹²

E: This statement is valid. Class CL1 implements the interface. This is because class CL1 implements the interface ifl using the INTERFACES statement in the public section of the class definition. The INTERFACES statement makes the class compatible with the interface and inherits all the components of the interface. The class must then provide an implementation for the interface method ml in the implementation part of the class, unless the method is declared as optional or abstract¹²

D: This statement is valid. In class CL2, the interface method is named ifl~ml. This is because class CL2 has a data member named m0_ifl of type REF TO ifl, which is a reference to the interface ifl. The interface ifl defines a method ml, which can be called using the reference variable m0_ifl. The interface method ml has the name ifl~ml in the class, where ifl is the name of the interface and the character ~ is the interface component selector¹²

The other statements are not valid, as they have syntax errors or logical errors. These statements are:

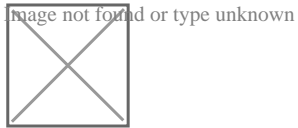
A: This statement is not valid. In class CL1, the interface method is named ifl~ml, not if-ml. This is because class CL1 implements the interface ifl using the INTERFACES statement in the public section of the class definition. The interface ifl defines a method ml, which can be called using the class name or a reference to the class. The interface method ml has the name ifl~ml in the class, where ifl is the name of the interface and the character ~ is the interface component selector. Using the character - instead of the character ~ will cause a syntax error¹²

B: This statement is not valid. Class CL2 does not use the interface, but only has a reference to the interface. This is because class CL2 has a data member named m0_ifl of type REF TO ifl, which is a reference to the interface ifl. The interface ifl defines a method ml, which can be called using the reference variable m0_ifl. However, class CL2 does not implement the interface ifl, nor does it inherit the interface components. Therefore, class CL2 does not use the interface, but only references the interface¹²

Question 8

Question Type: MultipleChoice

Refer to the Exhibit.



when you attempt to activate the definition, what will be the response?

Options:

- A) Activation error because the field names of the union do not match
- B) Activation error because the field types of the union do not match
- C) Activation error because the key fields of the union do not match
- D) Activation successful

Answer:

A

Explanation:

The response will be an activation error because the field names of the union do not match. This is because the field names of the union must match in order for the definition to be activated. The union operator combines the result sets of two or more queries into a single result set. The queries that are joined by the union operator must have the same number and type of fields, and the fields must have the same names¹. In the given code, the field names of the union do not match, because the first query has the fields carrname, connid, cityfrom, and cityto, while the second query has the fields carrname, carrier_id, cityfrom, and cityto. The field connid in the first query does not match the field carrier_id in the second query. Therefore, the definition cannot be activated.

To Get Premium Files for C_ABAPD_2309 Visit

https://www.p2pexams.com/products/c_abapd_2309

For More Free Questions Visit

<https://www.p2pexams.com/sap/pdf/c-abapd-2309>

