



MB-500^{Q&As}

Microsoft Dynamics 365 Finance and Operations Apps Developer

Pass Microsoft MB-500 Exam with 100% Guarantee

Free Download Real Questions & Answers **PDF** and **VCE** file from:

<https://www.geekcert.com/mb-500.html>

100% Passing Guarantee
100% Money Back Assurance

Following Questions and Answers are all new published by Microsoft
Official Exam Center

- ⚙️ **Instant Download** After Purchase
- ⚙️ **100% Money Back** Guarantee
- ⚙️ **365 Days** Free Update
- ⚙️ **800,000+** Satisfied Customers





QUESTION 1

You are adding a new field to the SalesTable form.

You must use an extension to add a status field onto the form.

You need to create the extension in the Application Object Tree (AOT) and add the extension to the demo Extensions model.

Solution: Navigate to the user interface forms node for the SalesTable form and modify the form.

Does the solution meet the goal?

A. Yes

B. No

Correct Answer: B

QUESTION 2

A company uses Dynamics 365 Finance.

You have an entity named VendVendorV2Entity. You add a new column to the entity.

You need to ensure that the new column is available in the entity for data transfer activity.

Which two actions should you perform? Each correct answer presents part of the solution.

NOTE: Each correct selection is worth one point.

A. Select the Generate mapping button on the Mapping details form.

B. Select the Validate button on the Entity list page.

C. Select the Entity structure button on the Entity list page.

D. Select the Modify target mapping button on the Entity list page.

Correct Answer: AB

Using the configuration key metadata built during entity refresh list, run time validations are performed in the following use cases.

1.

When user clicks Validate on the entity list.

2.

When the user generates mapping.

Reference: <https://docs.microsoft.com/en-us/dynamics365/fin-ops-core/dev-itpro/data-entities/data-entities>



QUESTION 3

You need to create a chain of command method for inserting the data on the table for the new fields. How should you complete the code? To answer, select the appropriate configuration in the answer area. NOTE: Each correct selection is worth one point.

Hot Area:

**Answer Area**

```
[ExtensionOf(TableStr(WHSTmpStatusChange))]  
[ExtensionOf(ClassStr(WHSTmpStatusChange))]  
  
class newWHSTmpStatusChange_Table_Extension  
{  
    final  
    public  
    {  
        public void insert()  
        {  
            EcoResProduct ecoResProduct;  
            EcoResProductTranslation ecoResProductT;  
            InventSum inventSum;  
            select firstonly InventBatchId from inventSum  
            where inventSum.ItemId == this.ItemId && inventSum.LicensePlateId == this.LicensePlateId  
            && inventSum.InventBatchId != "" && inventSum.PhysicalInvent > 0;  
  
            select firstonly Name from ecoResProductT  
            where EcoResProductT.LanguageId == CompanyInfo::languageId()  
            join EcoResProduct  
            where EcoResProduct.RecId == EcoResProductT.Product  
            && EcoResProduct.displayproductnumber == this.ItemId;  
  
            this.newEcoResProductName = ecoResProductT.Name;  
  
            next insert();  
            next public void insert()  
            Insert();  
  
            this.newInventBatchId = InventSum.InventBatchId;  
            this.newInventBatchId = InventSum.InventSiteId;  
            InventSum.InventBatchId = this.newInventBatchId;  
  
        }  
    }  
}
```

Correct Answer:

**Answer Area**

```
[ExtensionOf(TableStr(WHSTmpStatusChange))]  
[ExtensionOf(ClassStr(WHSTmpStatusChange))]  
  
class newWHSTmpStatusChange_Table_Extension  
{  
    final  
    public  
  
    {  
        public void insert()  
        {  
            EcoResProduct ecoResProduct;  
            EcoResProductTranslation ecoResProductT;  
            InventSum inventSum;  
            select firstly InventBatchId from inventSum  
            where inventSum.ItemId == this.ItemId && inventSum.LicensePlateId == this.LicensePlateId  
            && inventSum.InventBatchId != "" && inventSum.PhysicalInvent > 0;  
  
            select firstly Name from ecoResProductT  
            where EcoResProductT.LanguageId == CompanyInfo::languageId()  
            join EcoResProduct  
            where EcoResProduct.RecId == EcoResProductT.Product  
            && EcoResProduct.displayproductnumber == this.ItemId;  
  
            this.newEcoResProductName = ecoResProductT.Name;  
  
            next insert();  
            next public void insert()  
            Insert();  
  
            this.newInventBatchId = InventSum.InventBatchId;  
            this.newInventBatchId = InventSum.InventSiteId;  
            InventSum.InventBatchId = this.newInventBatchId;  
  
        }  
    }  
}
```

<https://learn.microsoft.com/en-us/dynamics365/fin-ops-core/dev-itpro/extensibility/addmethod-table>

QUESTION 4



A company uses Dynamics 365 Finance.

You implement an abstract class named Car.

The Car class has the following:

1.

A real member called weight that is used to store the weight of the car.

2.

An abstract getWeight method that returns the value of the weight member.

The Car class code is:

- ☐ A.

```
public int calc(int _value)
{
    if (_value > 10)
    {
        throw error('The value cannot be greater than 10.');
```
- ☐ B.

```
public int calc(int _value)
{
    if (_value > 10)
    {
        throw error('The value cannot be greater than 10.');
```
- ☐ C.

```
public int calc(int _value)
{
    int ret = next calc(_value);
    if (_value > 10)
    {
        throw error('The value cannot be greater than 10.');
```
- ☐ D.

```
public int calc(int _value)
{
    int ret;
    if (_value <= 10)
    {
        ret = next calc(_value);
    }
    return ret;
}
```

You must create a child class named BigCar.

You need to implement the Car class by using the BigCar child class.

Which code segment should you use?



- A. class BigCar extends Car { void new(real _weight) { weight = _weight; } public real getweight() { return weight; }}
- B. class BigCar implements Car { void new(real _weight) { weight = _weight; } public real getweight() { return weight; }}
- C. class BigCar extends car { public real getWeight() { return weight; }}
- D. class BigCar : car { void new(real _weight) { weight = _weight; } public real getweight() { return weight; }}

Correct Answer: C

QUESTION 5

After you answer a question in this section, you will NOT be able to return to it. As a result, these questions will not appear in the review screen.

You must create a new attribute class in which to mark other classes in the ISV solution and their respective methods with the error levels Warning and Error. You have the following class: (Line numbers are included for reference only.)

Name	Comments
AccountBase	Root EDT
AccountId	Derives from AccountBase

You need to ensure that the compiler will report an error if code calls the bike method.

Solution:

Create the following attribute class:

EDT name	Comment
CalendarID	root EDT
CalendarName	root EDT
BasicCalendarID	derives from CalendarID

Insert the following code at line 04:

```
[SysObsoleteAttribute("Deprecated. Please use the bicycle class instead.", false)]
```

Does the solution meet the goal?

A. Yes

B. No

Correct Answer: B

The second parameter of SysObsoleteAttribute should be true to produce a compiler error message. If it set to false it produces a compiler warning message.

SysObsoleteAttribute The system provides several attributes, including the SysObsoleteAttribute class. One use of the SysObsoleteAttribute class is to tell the X++ compiler that it should fail if a particular method is called in the source



code. The X++ compiler rejects the compile, and it displays the specific message that is stored in this use of the attribute. The SysObsoleteAttribute class can also be used to tell the compiler to issue warning messages instead of errors.

```
// X++
```

```
[SysObsoleteAttribute("The Automobile class might be a better choice for faster performance.", false)]
```

```
class Bicycle
```

```
{
```

```
// Members of the Bicycle class go here.
```

```
}
```

Note: Each object (Form, class, report, method) in AX has any parent class. That parent class can be an x++ class or a kernel class. Super() means calling a method with the same name in parent class.

Writing code before super() means, doing something before "execution"

Writing code after super() means doing something after "execution"

Reference: <http://axrachit.blogspot.com/2012/12/ax2012-understanding-attributes.html> <https://learn.microsoft.com/en-us/dynamicsax-2012/developer/overview-of-attribute-classes> <https://learn.microsoft.com/en-us/dynamics365/fin-ops-core/dev-itpro/dev-ref/xpp-inheritance>

[MB-500 PDF Dumps](#)

[MB-500 Practice Test](#)

[MB-500 Exam Questions](#)