

# DUMPS ARENA

## Adobe Certified Master - Magento Commerce Architect

Adobe AD0-E704

Version Demo

Total Demo Questions: 10

Total Premium Questions: 89

Buy Premium PDF

<https://dumpsarena.co>

[sales@dumpsarena.co](mailto:sales@dumpsarena.co)

[sales@dumpsarena.co](mailto:sales@dumpsarena.co)  
[dumpsarena.co](https://dumpsarena.co)

**QUESTION NO: 1**

You are integrating a Magento store with a 3rd-party OMS. The orders created on the Magento side must be uploaded to the OMS, there is no need to pull any data from the OMS into Magento. The OMS already has established RabbitMQ infrastructure so you decided to reuse it.

In addition to the implementation of the business logic for uploading orders, which two configurations must be done in your module to make the integration work?

- A. Publishers in `queue_publisher.xml`
- B. Consumers in `queue_consumer.xml`
- C. Topic declarations in `communication.xml`
- D. Routing rules in `queue_topology.xml`

**ANSWER: A B****QUESTION NO: 2**

You have a task to modify the grand total on the checkout page with a negative price adjustment which depends on the shipping address postcode. What are two approaches to be used in this case?

- A. Create a new shopping cart price rule with a condition based on the shipping address postcode
- B. Create a JavaScript mixin for the `Magento_Checkout/js/view/summary/subtotal`, change total there and Magento will submit the modified total when placing an order
- C. Create a new total collector which adds a negative price adjustment based on the postcode
- D. Create an observer on the event `sales_quote_collect_totals_after`, get the quote object, check the postcode and set the modified grand total into it

**ANSWER: A C****QUESTION NO: 3**

Suppose in a store there are 500 categories and it's increasing as per the need. Let's say your frontend developer asked you to modify the header and include CSS files on some specific category pages based on the custom category attribute selection from backend. What should be the acceptable and easiest way to do this, so that no new file creation and code update will be required after implementation:

- A. create a new layout XML file for the category called `catalog_category_view.xml` and add the layout modification code
- B. create a new layout XML file for the category called `catalog_category_view_selectable_CustomLayout.xml` and assign it to the needed categories from backend, and add the layout modification code in it.

- C. create a custom layout handle for the category using observer layout\_load\_before and add the layout modification code
- D. create a custom layout handle for the category using observer layout\_load\_after and add the layout modification code

**ANSWER: C**

#### QUESTION NO: 4

You are creating a module to enable customers to attach an image file to the quote items while adding a product to the cart. To save development time, you want to build on top of functionality provided by the class `\Magento\Catalog\Model\ImageUploader` as much as possible.

The class constructor signature is:

What do you do to process the uploaded file?

```
public function __construct(
    \Magento\MediaStorage\Helper\File\Storage\Database $coreFileStorageDatabase,
    \Magento\Framework\Filesystem $filesystem,
    \Magento\MediaStorage\Model\File\UploaderFactory $uploaderFactory,
    \Magento\Store\Model\StoreManagerInterface $storeManager,
    \Psr\Log\LoggerInterface $logger,
    $baseTmpPath,
    $basePath,
    $allowedExtensions
)
```

- A. Configure a plugin for the method `\Magento\Catalog\Model\ImageUploader::moveFileFromTmp()` in `di.xml`
- B. Create a custom implementation because the catalog imageUploader can not be used to for files associated with quote items
- C. Create a custom Class implementing `\Magento\Catalog\Model\ImageUploaderInterface` and add it to the `$uploaders` argument of the `UploaderPoolInterface` via `di.xml`
- D. Configure a Virtual type based on `\Magento\Catalog\Model\ImageUploader` Specifying `$baseTmpPath`, `$basePath` and `SallowedExtensions`

**ANSWER: D**

#### QUESTION NO: 5

You need to build a custom module to add a notice to the Advanced Inventory section on the product form. The notice text can be specified in the system configuration. How do you implement this?

- A. You add a plugin on the form `\Magento\Catalog\Ui\DataProvider\Product\Form\ProductDataProvider` to add a field in the Advanced Inventory section
- B. YOU add a plugin on `Magento\CatalogInventory\Ui\DataProvider\Product\Form\Modifier\AdvancedInventory` to modify the Section metadata
- C. You inject a custom block into the reference container in the `catalog_product_index` layout
- D. You add a field to `product_form.xml` as a child of the `stock_data` fieldset

**ANSWER: D****QUESTION NO: 6**

You need to create a Magento CLI command to return information from the system. What two steps are necessary?

- A. Add your command to the \Magento\Framework\Console\CommandListInterface using di.xml
- B. Create a cli.xml that will declare the command to run
- C. Implement \Magento\Framework\Console\CommandInterface to enable command auto-discovery
- D. Create your command class

**ANSWER: A D****QUESTION NO: 7**

While reviewing a module you found an upgrade script with the code:

```
/** @var Magento\Sales\Setup\SalesSetup */  
$salesSetup->addAttribute(  
    \Magento\Sales\Model\Order::ENTITY,  
    'custom_attribute',  
    $properties  
);
```

What two actions will be done by the SalesSetup::addAttribute method?

- A. The custom\_attribute added to the eav\_attribute table
- B. The custom\_attribute IS added to the order\_eav\_attribute table
- C. The sales\_order\_grid table is altered with a new field custom\_attribute
- D. The sales\_order table is altered with a new field custom\_attribute

**ANSWER: B D****QUESTION NO: 8**

You have created a custom block MyCompany MyModule/Block/Product/Update/ php. Which two approaches allow you to set a template to your block?

- A. Create protected variable \$\_template
- B. Create template view /frontend/template/product/update, phtml
- C. Configure block's constructor \$template parameter via di.xml
- D. Add an argument node in layout.xml: ...

**ANSWER: B C**

**QUESTION NO: 9**

You are starting a project and realized you need to clean the content of a container. How do you achieve this?

- A.** You create a view/layout/mycompany\_mymodule\_index.xml and add a referenceContainer. You will then remove the existing blocks one by one
- B.** YOU create a view/layout/mycompany\_mymodule\_index.xml With the referenceContainer and remove all the blocks by adding `remove name='*' />`
- C.** The layout merging logic will override a container if you reference the container and load your module after the original one using sequence in your module.xml
- D.** YOU create a view/layout/override/mycoapany\_mymodule\_index.xml and add the new blocks to the referenceContainer to override the previous ones

**ANSWER: D**

**QUESTION NO: 10**

A furniture merchant have 100k products in the store. In a custom product list page, he advised you to enable the caching as per user paginate the products. So what should the approach to cache the block data as per pagination request:

- A.** In block class toHtml() method, define cache\_lifetime, cache\_tags and cache\_key\_
- B.** In block class \_construct() method, define cache\_lifetime, cache\_tags and cache\_key\_
- C.** In block class toHtml() method, set \_isScopePrivate property to false as per requested page number.
- D.** In block class \_construct() method, set \_isScopePrivate property to false as per requested page number.

**ANSWER: B**