# Question 1

The Job_Application__c custom object has a field that is a Master-Detail relationship to the Contact object, where the Contact object is the Master. As part of a feature implementation, a developer needs to retrieve a list containing all Contact records where the related Account Industry is 'Technology' while also retrieving the contact's Job_Application__c records.

Based on the object's relationships, what is the most efficient statement to retrieve the list of contacts?

## Options:

**A-** [SELECT Id, (SELECT Id FROM Job_Applications_r) FROM Contact WHERE Account.Industry = 'Technology'];

**B-** [SELECT Id, (SELECT Id FROM Job_Applications_r) FROM Contact WHERE Accounts.Industry = 'Technology'];

**C-** [SELECT Id, (SELECT Id FROM Job_Applications_c) FROM Contact WHERE Accounts.Industry = 'Technology'];

**D-** [SELECT Id, (SELECT Id FROM Job_Application_c) FROM Contact WHERE Account.Industry = 'Technology'];

## Answer:

B

# Question 2

Universal Containers has a Visualforce page that displays a table of every Container_c. being ....... Is falling with a view state limit because some of the customers rent over 10,000 containers.

What should a developer change about the Visualforce page to help with the page load errors?

## Options:

**A-** Use Lazy loading and a transient List variable.

**B-** Use JavaScript remoting with SOQL Offset.

**C-** Implement pagination with an OffsetController.

**D-** Implement pagination with a StandardSetController,

## Answer:

D

# Question 3

A developer created a Lightning web component called statusComponent to be inserted into the Account record page.

Which two things should the developer do to make the component available?

## Options:

**A-** Add <isExposed> true</isExposed> to the statusComponent.js-meta ml file.

**B-** Add <target> lighting _RecordPage </target> to the statusComponent.js-meta ml file.

**C-** Add < masterLabel>Account</master Label> to the statusComponent.js-meta ml file.

**D-** Add<target> Lightning_RecordPage </target> to the statusComponent.js file.

## Answer:

A, B

# Question 4

A develop completed modification to a customized feature that is comprised of two elements:

Apex trigger

Trigger handler Apex class

What are two factors that the developer must take into account to properly deploy the modification to the production environment?

## Options:

**A-** Apex classes must have at least 75% code coverage org-wide.

**B-** At least one line of code must be executed for the Apex trigger.

**C-** All methods in the test classes must use @isTest.

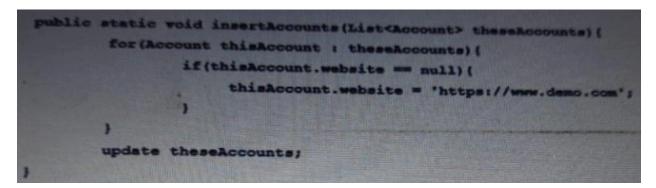**D-** Test methods must be declared with the testMethod keyword.

## Answer:

A, B

# Question 5

**Question Type:** **MultipleChoice**

Considering the following code snippet:

```
public static void insertAccounts(List<Account> theseAccounts){
    for(Account thisAccount : theseAccounts){
        if(thisAccount.website == null){
            thisAccount.website = 'https://www.demo.com';
        }
    }
    update theseAccounts;
}
```

When the code executes a DML exception is thrown.

How should the developer modify the code to ensure exceptions are handled gracefully?

## Options:

A- Implement a try/catch block for the DML.

B- Remove null items from the list if Accounts.

C- Implement the upsert DML statement.

D- Implement Change Data Capture

## Answer:

A

# Question 6

A developer is creating a Lightning web component to show a list of sales records.

The Sales Representative user should be able to see the commission-field on each record. The Sales Assistance user should be able to see all field on the record except the commission field.

How should this be enforced so that the component works for both users without showing any errors?

## Options:

**A-** Use Lightning Data Service to get the collection of sales records.

**B-** Use Lightning Locker Service to enforce sharing rules and field-level security.

**C-** Use with SECURITY_EMFoRCED in the SOQL that fetches the data for the component.

**D-** Use security. stripInaccessible to remove fields inaccessible to the current user.

## Answer:

D

# Question 7

A third-party vendor created an unmanaged Lightning web component. The Salesforce Administrator wishes to expose the component only on Record Page Layouts.

Which two actions should the developer take to accomplish this business objective?

Choose 2 answers

## Options:

**A-** Specify lightningCommunity_Page as a target in the XML file.

**B-** Ensure isExposed is set to true on the XML file.

**C-** Specify lightningCommunity_Page_Layout as a target in the XML file.

**D-** Specify lightning_RecordPage as a target in the XML file.

## Answer:

B, D

# Question 8

A Salesforce Administrator is creating a record-triggered flow. When certain criteria are met, the flow must call an Apex method to execute complex validation involving several types of objects.

When creating the Apex method, which annotation should a developer use to ensure the method

Can be used within the flow?

## Options:

**A-** @future

**B-** @RemoteAction

**C-** @InvocableMethod

**D-** @AuraEnaled

## Answer:

C

# Question 9

Universal Container uses Salesforce to create orders.

When an order is created, it needs to sync with the-in-house order fulfillment system. The order fulfillment system can accept SOAP messages over the HTTPS. If the connection fails, messages should be retried for up to 24 hours.
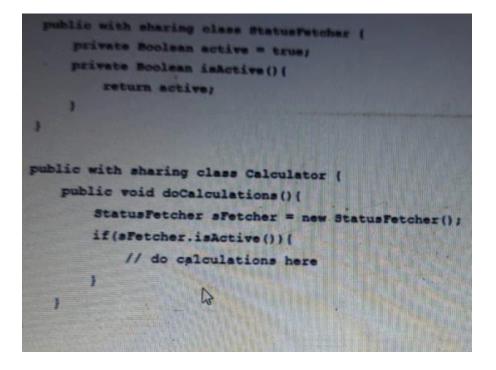
What is the recommended approach to sync the orders in Salesforce with the order fulfillment system?

## Options:

**A-** Set up a Workflow Rule outbound message to the order fulfillment system.

**B-** Create an after insert trigger on the Order object to make a callout to the order fulfilment system

**C-** Write an Apex SOAP service to integrate with the order fulfillment system.

**D-** Use Process Bulkier to call an invocable Apex method that sends a message to the order fulfilment system.

## Answer:

B

# Question 10

A developer wrote the following two classes:

```
public with sharing class StatusFetcher {
    private Boolean active = true;
    private Boolean isActive(){
        return active;
    }
}

public with sharing class Calculator {
    public void doCalculations(){
        StatusFetcher sFetcher = new StatusFetcher();
        if(sFetcher.isActive()){
            // do calculations here
        }
    }
}
```

The StatusFetcher class successfully compiled and saved. However, the Calculator class has a compile time error.

How should the developer fix this code?

## Options:

**A-** Change the class declaration for the statusFetcher class to public with inherited sharing.

**B-** Make the is Active method in the StatusFetcher class public.

**C-** Make the doCalculations method in the Calculation class private.

**D-** Change the class declaration for the Calculator class to public with inherited sharing.

## Answer:

B