A developer must write an Apex method that will be called from a Lightning component. The method may delete an Account stored in the accountRec variable. Which method should the developer use to ensure **only users with the correct permissions** can successfully delete Accounts?

**A.**

public static void deleteAccount(Account accountRec) {

delete accountRec;

}

**B.**

@AuraEnabled

public static void deleteAccount(Id accountId) {

Account acc = [SELECT Id FROM Account WHERE Id = :accountId];

Database.delete(acc, false);

}

**C.**

@AuraEnabled

public static void deleteAccount(Account accountRec) {

if (Schema.sObjectType.Account.isDeletable()) {

delete accountRec;

}

}

**D.**

@AuraEnabled

public static void deleteAccount(Id accountId) {

if (UserInfo.getProfileId() == 'System Administrator') {

delete [SELECT Id FROM Account WHERE Id = :accountId];

}

}

Answer :c

A developer is writing an Apex method to fetch Accounts for use in a Lightning component. The method must be accessible to the Lightning component. Which annotation must the developer use?

**A.** @RemoteAction  
**B.** @AuraEnabled  
**C.** @InvocableMethod  
**D.** @WebService

Answer : B

A developer writes the following Apex method:

public with sharing class AccountController {

@AuraEnabled

public static List<Account> getAccounts() {

return [SELECT Id, Name, Industry FROM Account LIMIT 10];

}

}

What is the risk in this code when called from a Lightning component?

**A.** Query may hit governor limits.  
**B.** Method is not bulkified.  
**C.** Field-level security is not enforced.  
**D.** Sharing rules are not applied.

Answer : C

Which statement about using @AuraEnabled methods in Apex is **true**?

**A.** They can only return String values.  
**B.** They cannot be static.  
**C.** They must be declared as public or global.  
**D.** They do not support custom Apex classes as return types.

**Answer: C**

A developer writes this method:

public with sharing class OpportunityController {

@AuraEnabled(cacheable=true)

public static List<Opportunity> getOpenOpps() {

return [SELECT Id, Name, StageName FROM Opportunity WHERE IsClosed = false];

}

}

Why is cacheable=true important here?

**A.** It allows method execution in batch context.  
**B.** It makes the method read-only and enables client-side caching.  
**C.** It allows the method to be called from multiple namespaces.  
**D.** It enforces CRUD and FLS checks automatically.

Answer: C

A developer needs to make a callout to an external REST service when an Opportunity is updated. Which Apex feature should be used?

**A.** @AuraEnabled method  
**B.** @future(callout=true) method  
**C.** Database.Batchable class  
**D.** @InvocableMethod

**Answer: B**

Requirements state that a **child record is deleted when its parent is deleted**, and a **child can be moved to a different parent when necessary**. Which type of relationship should be built between the parent and child objects in Schema Builder to support these requirements?

**A.** Lookup Relationship  
**B.** Master-Detail Relationship  
**C.** Hierarchical Relationship  
**D.** Many-to-Many Relationship

Answer : B

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?

**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 : D**

Which two events need to happen when deploying to a production org? (Choose two.)

1. **All Process Builder Processes must have at least 1% test coverage.**
2. **All Apex code must have at least 75% test coverage.**
3. **All triggers must have at least 1% test coverage.**
4. **D. All Visual Flows must have at least 1% test coverage.**

**Answer : B,D**

How should a developer avoid hitting the governor limits in test methods?

**A. Use @TestVisible on methods that create records.**

**B. Use Test.loadData() to load data from a static resource.**

**C. Use @IsTest (SeeAllData=true) to use existing data.**

**D. Use Test.startTest() to reset governor limits.**

**Answer : D**

An org has a single account named `˜NoContacts' that has no related contacts. Given the query:List<Account> accounts = [Select ID, (Select ID, Name from Contacts) from Account where Name=`˜NoContacts'];What is the result of running this Apex?

**A. accounts[0].contacts is invalid Apex.**

**B. accounts[0].contacts is an empty Apex.**

**C. accounts[0].contacts is Null.**

**D. A QueryException is thrown.**

**Answer: B**