1. A helper class is created to be used in Trigger, batch class and a custom controller. How will you make sure that only specific code will executed in Trigger context?

**A. Use trigger context variable Trigger.isExecuting**

B. Check if System.context == 'Trigger'

C. user Trigger.isRunning() function

2. What will be the error in following trigger -

Trigger on Lead (Before update){

for(Lead ld : Trigger.new){

if(lead.Email != Trigger.oldMap.get(ld.Id).Email){

lead.custom\_email\_\_c = lead.Email;

update lead;

}

}

}

A. Trigger goes in recursion

B. DML limit excetion

**C. DML exception**

D. NULL Pointer exception

3. What is the output of following code -

Integer a = 0;

Integer b;

a = a++;

a++;

Integer c = a\*5;

String flag = String.valueOf(c)+1;

System.debug(flag+';'+b);

**A) 51;null**

b) 101;null

c) 51;0

d) 101;0

4. What is the output of following code

Integer a = 0;

Integer b = 1;

do{

a++; b++;

system.debug('a : '+a);

}while(a == 80 && b == 81);

a) a : 81

**b) a: 1**

c) It will print 'a : 1' upto 'a : 81'

d) Compiler error

5) Given 3 objects - Movie, Actor, Contract. Each Movie can have multiple actors and each actor can have multiple movies.

Every contract keeps the track of which actor plays a role in which Movie.

Also, there are different owner of different contracts and they want to be able to see the contracts irrespective of whether they can

access a Movie or an Actor record.

How will you design the data model?

a) Master - Detail relationship on Contract for Movie and Lookup relationship on Contract for Actor

b) Lookup on Contract for Movie and Master - Detail relationship on Contract for Actor

c) Master - Detail relationship on Contract for Movie and Master - Detail relationship on Contract for Actor

**d) Lookup relationship on Contract for Movie and Lookup relationship on Contract for Actor**

6) How will you determine the uniqueness of the objects added into a set?

**a) Provide equals and hashCode methods in your class**

b) Set handles the uniqueness itself

c) Check if the object is already exist in the set

7) A visualforce page uses Account standard controller and also has an extension that have save() method.

A button on the page calls the save action when clicked. Which action will be executed?

a) save action of Standard Controller

**b) save action of extension**

c) both the save actions of Standard Controller and extension

d) Compiler error due to same name

8) You are asked to override the new button of Lead object. What will you make sure while creating a visualforce page?

A**) Set Account object as the standard controller**

9) Which of the following can be used to get the list of fields in Salesforce of Account obejct -

**A) Schema.getGlobalDescribe().get('Account').getDescribe().fields**

**B) Schema.SObjectType.Account.fields**

10) Which tool you can use to post a message to chatter when lead status is changed -

1) Field history tracking

**2) Process Builder**

3) Workflow Action

12) How will you search Phone in Accounts, Contacts and Leads

A) SOSL with with ALL ROWS

B) SOQL

**C) SOSL**

D) SOQL with with ALL ROWS

13) SOSL query to find phone number in Account, Leadf and Contact -

**A) SELECT Name, Max(CreatedDate) FROM Account ORDER BY Name**

14) How will you migrate components from a developer org to unrelated production org?

**A) Using Force.com IDE**

**B) Creating an Unmanaged package and installing it in the production org**

15) Which of the following can be used to deploy destructive changes -

**A) Force.com IDE**

**B) Workbench**

C) Change sets

D) Tooling Api

16) How to change the debug level of debug log in Developer console

**A) Debug-> Change log levels**

17) Which primitive data type is used to store number field in Salesforce ===IMP

A) Number

B) Integer

**C) Decimal**

18) How will you get value of multi-select picklist in apex?

A) Set<String>

B) List<String>

C) Comma separated values in a String

**D) Semicolon separated values in a String**

19) Which of the following can be used to run the test classes?

**A) Setup->Apex Test Execution**

**B) Developer Console**

**C) Tooling Api**

D) Metadata Api

20) When does the code coverage of an org is calculated?

**A) After running unit tests**

B) After Deployment Validation

20) Can we create rollup on formula field?

**A) Yes, bt the formula shouldn't contain non-deterministic fields**

B) No

C) Yes

21) To check why a unit test was failing with exception, the developer copied the code and pasted in anononymouse window. The code was running without exception.

What would be the reason?

**a) The unit test relied on org data**

b) The unit test used SeeAllData = true

c) The unit test declaration was incorrect

22) An object already has 2 triggers. What should the developer must consider regarding the order of execution of trigger? ===IMP

**a) Order of execution cannot be determined**

b) Order of execution is based on the creation of trigger

c) Both the triggers can run in parallel

d) Need to specify which trigger has to be covered

23) A developer runs the following anonymous code block:

List<Account> acc = [SELECT Id FROM Account LIMIT 10;];

Delete acc;

Database.emptyRecycleBin(acc);

system.debug(Limits.getDMLStatements() + ‘, ‘+Limits.getLimitsDMLStatements());

What is the result?

**a. 2, 150**

b. 150, 2

c. 11, 150

d. 150, 11

24) Which of the following stands as Model in MVC pattern?

**1) Custom Fields**

2) Javascript

3) Apex classes

25) What is a valid statement about Apex classes and interfaces? (2) ===IMP

a. A class can have multiple levels of inner classes.

b. **Exception classes must end with the word exception.**

c. The default modifier for an interface is private.

d. **The default modifier for a class is private.**

26) How to avoid view state ? ===IMP

**a) Transient keyword**

**b) javascript remoting**

27) Which of the following can be used to migrate metadata from one sandbox to another ===IMP

**a) Metadata Api**

**b) force.com IDE**

**c) Change set**

d) Data Loader

28) Which of the two data type in apex can be sued to hold Id of record dynamically? ===IMP

**a) String**

**b) Object**

29) Which data type in apex used to hold key value pair? ===IMP

**a) Map**

b) List

c) Object

d) sObject

30) Ways to iterate map -===IMP

**a) for(Account a : accMap.values() ){}**

**b) for(Id accId : accmap.keySet()){}**

**c) for(Integer i=0; i< accmap.size(); i++){}**

31) Ways to iterate over list - ===IMP

**a) for(Account a : accList ){}**

**b) for(Integer i=0; i< accList.size(); i++){}**

32) List<Account> accList = [Select Id,(select Id,Name from Contact) from Account ]; ===IMP

What are the possible exceptions we can get with the above query ? Choose 2

1. soql limit error on contact records

2. soql limit error on account records

3. too many soql queries

4. malformed query - soql exception

33) which 2 below will compile ? ===IMP

1.Integer i=3.14159;

2.Double d = 3.14159;

3. Decimal d = 3.14159;

4. Long l = 3.14159;

34) List<Account> accList = [Select Id,(select Id,Name from Contact) from Account ]; ===IMP

If there is only one account in the system and there is no contact associated to it. Which of the below is correct ?

1.too many soql quries

2.accList[0] is null

3.accList[0].contacts is empty list

4.List has no rows for assignment

35) Which belongs to controller part of MVC framework ? ===IMP

1.PRocess builder

2.WF Rules

3.Standard Object

4.Fields

36) What are the different ways to run test class ?(Choose 3) ===IMP

1.Tooling API

2.SFDC Setup

3.Workbench

4.MetadataAPI

5.DevConsole

37) how to convet List<sObject> to the account and contact list ?

1.Get the ids of each rec and identify frm the first 3 digits

2.getSobjectType

3.getSObjectName

4.use try catch to get to know the objects.

38) significance of after insert triggers

39) significance of <ltng:require>===IMP

Loads scripts and stylesheets while maintaining dependency order.

The styles are loaded in the order that they are listed.

The styles only load once if they are specified in multiple <ltng:require> tags in the same component or across different components.

40) a custom object machinery has records like Cranes,Tractor, and other tools. A sales manager can associate one or more tools to a construction object

1.Lookup on machinery to construction

2.masterdetail on construction to machonery

3. Junction object to hold both these values

4. lookup on construction to machinery

41) Different ways of iterating the list -List<Account> accList?

1. for(Account acc:accList){}

2. for (Integer i=0;i<accList.size();i++){}

3. for(List ls;accList){}

4. for (accList){}

42) Different ways of iterating the map - Map<Id,Account> mapSet ?

1.for(Id ids:mapSet.keySet()){}

2.for(Account acs:mapSet.values()){}

3.for(Account act:mapSet.keySet()){}

43) Style Sheets===IMP

1. apex:stylesheet with static resource

2. inline CSS code

3. standardStylesheets

4.<apex:slds> to include lightning stylesheet

44) What are the languages supported in Heroku===IMP

45) Deploy Heroku app to production

46) How do you support languages that are not supported in Salesforce

47) Before moving app to production, what are things (Dynos)

48) Ltng : require

49) Refence key

a. Trigger

b. Process Flow

50) Deploy Visual force

Create test Class for Controller

51) Recruit and Managers – Visual force page

Controller with Sharing

52) Schema.getDescribe

RecordType

PickList

53) Test Setup

Test Insert Account

Zero

54) Test.Start – Refresh Governor Limits===IMP

55) Extends Exception

56) Lightning Component - Description – Text ===IMP

a> Interface

b> Helper

c> Extends

57) Although custom controllers and controller extension classes execute in **system mode** and thereby ignore user permissions and field-level security, you can choose whether they respect a user's organization-wide defaults, role hierarchy, and sharing rules by using the **with sharing** keywords in the class definition.

58) A custom controller is an Apex class that implements all the logic for a page **without leveraging a standard controller**

**59)** A controller extension is an Apex class that extends the functionality of a standard or custom controller

1. You want to leverage the built-in functionality of a standard controller but override one or more actions, such as edit, view, save, or delete.
2. You want to add new actions.
3. You want to build a Visualforce page that respects user permissions

60) **A custom controller** is an Apex class that uses the default, **no-argument constructor** for the outer, top-level class. **You cannot** create a custom controller constructor that includes parameters.

61) **A controller extension** is any Apex class containing a constructor that takes a single argument of type ApexPages.StandardController or CustomControllerName, where CustomControllerName is the name of a custom controller you want to extend.

**62)** A **custom list controller** is similar to a standard list controller. Custom list controllers can implement Apex logic that you define to show or **act on a set of records.**

**63)** Use the transient keyword to declare instance variables that can't be saved, and shouldn't be transmitted as part of the view state for a Visualforce page. For example:

Declaring variables as transient reduces view state size.

64) Since the view state is linked to form data, the View State tab only appears if your page contains an <apex:form> tag. In addition, the View State tab displays only on pages using [custom controllers or controller extensions](https://developer.salesforce.com/docs/atlas.en-us.pages.meta/pages/pages_controller_def.htm#pages_controller_def).

65) With the Process Builder, you can execute a wide range of actions, including:

* Create a record
* Update any related record
* Use a quick action to create a record, update a record, or log a call
* Launch a flow—flows are another automation tool ideal for complex branching logic
* Send an email
* Post to Chatter
* Submit for Approval

66) Lookup

Create a lookup relationship on Job Application that reference the Position object

Position – Parent

Job Application – Child

Create a lookup relationship on Job Application that reference the Candidate Object

Candidate – Parent

Job Application – Child

Create a lookup relationship Hiring Manager field on the Position object that reference the Standard User object

Standard User – Parent

Position - Child

67) Master Detail

Create a Master detail relationship on Review object that references the Job Application.

When a Job application is deleted, delete the Review feedback object as well.

Job Application – Parent

Review - Child

69) Roll up Summary - Total rating

The good news is that we can! A simple roll-up summary field on the Job Application object can summarize data from a set of related detail records and automatically display the output on a master record. Use roll-up summary fields to display the sum, minimum, or maximum value of a field in a related list, or the record count of all records listed in a related list.

70) Many to Many (Junction)

Create a Job Posting object to create many to many relationships between position and employment web sites.

Parent 1 – Position

Parent 2 – Employment web sites

Junction object – Job Posting

<https://developer.salesforce.com/docs/atlas.en-us.fundamentals.meta/fundamentals/adg_securing_data_access.htm>

71) Required permission for the Recruiter – All permissions

Required permission for the Hiring Manager - able to update the status of those job applications to specify which candidates should be selected or rejected.

Required permission for the Interviewer – FLS on position (Salary)

Required permission for the Standard Employee

72) Developer and Developer Pro sandboxes

Development and QA

73) Partial Copy

Integration

Batch

Training

UAT

74) Full Copy

Performance tuning

Staging

75) Use Apex if you want to:

* Create Web services.
* Create email services.
* Perform complex validation over multiple objects.
* Create complex business processes that are not supported by workflow.
* Create custom transactional logic (logic that occurs over the entire transaction, not just with a single record or object).
* Attach custom logic to another operation, such as saving a record, so that it occurs whenever the operation is executed, regardless of whether it originates in the user interface, a Visualforce page, or from SOAP API.

76) You can’t modify Apex using the Salesforce user interface in a Salesforce production org.

## 77) Developer Console

Writing code

Compiling code

Debugging

Testing

Checking performance

SOQL queries

Color coding and auto complete

77) At least 75% of your Apex code must be covered by unit tests

* Calls to System.debug are not counted as part of Apex code coverage.
* Test methods and test classes are not counted as part of Apex code coverage.

including positive and negative cases, as well as bulk and single records.

* Every trigger must have some test coverage.
* All classes and triggers must compile successfully.

78) Deploying Apex to Sandbox

The Force.com IDE and the Ant Migration Tool are free resources provided by Salesforce

79) Deploying Apex

Change Sets

Force.COM IDE

Ant migration tool

SOAP API

79) Currency fields are automatically assigned the type Decimal.

The Currency.newInstance

The Currency.newInstance static method creates a literal of type Currency

80) A common pitfall is to assume that an uninitialized boolean variable is initialized to false by the system. This isn’t the case. Like all other variables, boolean variables are null if not assigned a value explicitly.

## 81) Variable Scope

Variables can be defined at any point in a block ,  and take on scope from that point forward.

Sub-blocks can’t redefine a variable name that has already been used in a parent block

parallel blocks can reuse a variable name

82) Constants

Apex constants are variables whose values don’t change after being initialized once. Constants can be defined using the final keyword.

The final keyword means that the variable can be assigned at most once, either in the declaration itself, or with a static initializer method if the constant is defined in a class.

83) SOQL statements evaluate to a list of sObjects, a single sObject, or an Integer for count method queries.

84) Custom Index Considerations for Selective SOQL Queries

* The following fields are indexed by default.
  + Primary keys (Id, Name, and OwnerId fields)
  + Foreign keys (lookup or master-detail relationship fields)
  + Audit dates (CreatedDate and SystemModstamp fields)
  + RecordType fields (indexed for all standard objects that feature them)
  + Custom fields that are marked as External ID or Unique

# 85) Understanding Foreign Key and Parent-Child Relationship SOQL Queries

This works

for (Account a : [SELECT Id, Name, (SELECT LastName FROM Contacts)

FROM Account

WHERE Name = 'Acme']) {

Contact[] cons = a.Contacts;

}

**//The following example also works because we limit to only 1 contact**

for (Account a : [SELECT Id, Name, (SELECT LastName FROM Contacts LIMIT 1)

**FROM Account**

**WHERE Name = 'testAgg'])** {

Contact c = a.Contacts;

}

Causes an Error - You might get a QueryException in a SOQL for loop with the message Aggregate query has too many rows for direct assignment, use FOR loop.

for (Account acct : [SELECT Id, Name, (SELECT Id, Name FROM Contacts)

FROM Account WHERE Id IN **('<ID value>')**]) {

List<Contact> contactList = acct.Contacts; // Causes an error

Integer count = acct.Contacts.size(); // Causes an error

}

This works! To avoid getting this exception, use a for loop to iterate over the child records, as follows.

for (Account acct : [SELECT Id, Name, (SELECT Id, Name FROM Contacts)

FROM Account WHERE Id IN ('<ID value>')]) {

Integer count=0;

**for (Contact c : acct.Contacts) {**

**count++;**

**}**

}

86> getGlobalDescribe() – schema.SojbectResult <Strin, sOjectType)

Sobject sobj = getDescribe() -> getsOjbectTpe()

Field fld = getDecsribe() -> getsObjectField()

# 87> Custom Settings

List Custom Settings - Data in list settings does not vary with profile or user, but is available organization-wide

Hierarchy Custom Settings - A type of custom setting that uses a built-in hierarchical logic that lets you “personalize” settings for specific profiles or users. The hierarchy logic checks the organization, profile, and user settings for the current user and returns the most specific, or “lowest,” value

88) Anonymous blocks (Author Apex permission)

Developer console

Force.COM IDE

Execute anonymous SOAP API Call

89) The return result for anonymous blocks includes:

Status information for the compile and execute phases of the call

Debug log content, output

Apex Stack Trace if any exception with class name. method and line number

90) upsert triggers fire both before and after insert or before and after update triggers as appropriate.

91) merge triggers fire both before and after delete for the losing records, and both before and after update triggers for the winning record.

92) Be aware of the following considerations for trigger context variables:

* trigger.new and trigger.old cannot be used in Apex DML operations.
* You can use an object to change its own field values using trigger.new, but only in before triggers.
* In all after triggers, trigger.new is not saved, so a runtime exception is thrown.
* trigger.old is always read-only.
* You cannot delete trigger.new.

93) To view a debug log, from Setup, enter Debug Logs in the Quick Find box, then select **Debug Logs**. Then click **View** next to the debug log that you want to examine. Click **Download** to download the log as an XML file.

Log lines – time stamp and event identifier

94) To create your custom exception class, extend the built-in Exception class and make sure your class name ends with the word Exception, such as “MyException” or “PurchaseException”.

95) Heroku Languae support===IMP

Node

Ruby

Python and PHP

Java

Scala

Clojure

Kotlin

Go

* 96) If a test class contains a static member variable, and the variable’s value is changed in a testSetup or test method, the new value isn’t preserved. Other test methods in this class get the original value of the static member variable. This behavior also applies when the static member variable is defined in another class and accessed in test methods.
* 97) You can define the behavior of stub objects, which are created at runtime as anonymous subclasses of Apex classes. The stub API comprises the System.StubProvider interface and the System.Test.createStub() method.