

Android Developer Job Interview Questions And Answers



Interview Questions Answers

<https://interviewquestionsanswers.org/>

About Interview Questions Answers

Interview Questions Answers . ORG is an interview preparation guide of thousands of Job Interview Questions And Answers, Job Interviews are always stressful even for job seekers who have gone on countless interviews. The best way to reduce the stress is to be prepared for your job interview. Take the time to review the standard interview questions you will most likely be asked. These interview questions and answers on Android Developer will help you strengthen your technical skills, prepare for the interviews and quickly revise the concepts.

If you find any **question or answer** is incorrect or incomplete then you can **submit your question or answer** directly with out any registration or login at our website. You just need to visit [Android Developer Interview Questions And Answers](#) to add your answer click on the *Submit Your Answer* links on the website; with each question to post your answer, if you want to ask any question then you will have a link *Submit Your Question*; that's will add your question in Android Developer category. To ensure quality, each submission is checked by our team, before it becomes live. This [Android Developer Interview preparation PDF](#) was generated at **Wednesday 29th November, 2023**

You can follow us on FaceBook for latest Jobs, Updates and other interviews material.
www.facebook.com/InterviewQuestionsAnswers.Org

Follow us on Twitter for latest Jobs and interview preparation guides.
<https://twitter.com/InterviewQA>

If you need any further assistance or have queries regarding this document or its material or any of other inquiry, please do not hesitate to contact us.

Best Of Luck.

Interview Questions Answers.ORG Team
<https://InterviewQuestionsAnswers.ORG/Support@InterviewQuestionsAnswers.ORG>



Android Developer Interview Questions And Answers Guide.

Question - 1:

What is foreground Service?

Ans:

A foreground service performs some operation that is noticeable to the user. For example, we can use a foreground service to play an audio track. A Notification must be displayed to the user.

[View All Answers](#)

Question - 2:

What is bound Service?

Ans:

A service is bound when an application component binds to it by calling `bindService()`. A bound service offers a client-server interface that allows components to interact with the service, send requests, receive results. A bound service runs only as long as another application component is bound to it.

[View All Answers](#)

Question - 3:

Please explain what is a Pending Intent?

Ans:

If you want someone to perform any Intent operation at future point of time on behalf of you, then we will use Pending Intent.

[View All Answers](#)

Question - 4:

What is background Service?

Ans:

A background service performs an operation that isn't directly noticed by the user. In Android API level 26 and above, there are restrictions to using background services and it is recommended to use `WorkManager` in these cases.

[View All Answers](#)

Question - 5:

Explain what is application Context?

Ans:

This context is tied to the lifecycle of an application. The application context can be used where you need a context whose lifecycle is separate from the current context or when you are passing a context beyond the scope of an activity.

[View All Answers](#)

Question - 6:

Tell us what is DDMS? Describe some of its capabilities?

Ans:

DDMS is the Dalvik Debug Monitor Server that ships with Android. It provides a wide array of debugging features including:

- * port-forwarding services
- * screen capture
- * thread and heap information
- * network traffic tracking
- * incoming call and SMS spoofing
- * simulating network state, speed, and latency
- * location data spoofing

[View All Answers](#)

**Question - 7:**

Explain me what is Armv7?

Ans:

There are 3 CPU architectures in Android. ARMv7 is the most common as it is optimised for battery consumption. ARM64 is an evolved version of that that supports 64-bit processing for more powerful computing. ARMx86, is the least used for these three, since it is not battery friendly. It is more powerful than the other two.

[View All Answers](#)

Question - 8:

What is difference between Activity & Service?

Ans:

Activities are basically containers or windows to the user interface. Services is a component that is used to perform operations on the background. It does not have an UI.

[View All Answers](#)

Question - 9:

Do you know what is ANR, and why does it happen?

Ans:

'ANR' in Android is 'Application Not Responding.' It means when the user is interacting with the activity, and the activity is in the onResume() method, a dialog appears displaying "application not responding."

It happens because we start a heavy and long running task like downloading data in the main UI thread. The solution of the problem is to start your heavy tasks in the background using Async Task class.

[View All Answers](#)

Question - 10:

Do you know what are Handlers?

Ans:

Handlers are objects for managing threads. It receives messages and writes code on how to handle the message. They run outside of the activity's lifecycle, so they need to be cleaned up properly or else you will have thread leaks.

Handlers allow communicating between the background thread and the main thread.

A Handler class is preferred when we need to perform a background task repeatedly after every x seconds/minutes.

[View All Answers](#)

Question - 11:

What is the build process in Android?

Ans:

* First step involves compiling the resources folder (/res) using the aapt (android asset packaging tool) tool. These are compiled to a single class file called R.java. This is a class that just contains constants.

* Second step involves the java source code being compiled to .class files by javac, and then the class files are converted to Dalvik bytecode by the "dx" tool, which is included in the sdk 'tools'. The output is classes.dex.

* The final step involves the android apkbuilder which takes all the input and builds the apk (android packaging key) file.

[View All Answers](#)

Question - 12:

Explain me what is Android SDK?

Ans:

To develop a mobile application, Android developers require some tools and this requirement is satisfied by "Android SDK" which is a set of tools that are used for developing or writing apps.

It has a Graphical User Interface which emulates the Android environment. This emulator acts as an actual mobile device on which the developers write their code and then debug/test the same code to check if anything is wrong.

[View All Answers](#)

Question - 13:

Tell us how would you update the UI of an activity from a background service?

Ans:

We need to register a LocalBroadcastReceiver in the activity. And send a broadcast with the data using intents from the background service. As long as the activity is in the foreground, the UI will be updated from the background. Ensure to unregister the broadcast receiver in the onStop() method of the activity to avoid memory leaks. We can also register a Handler and pass data using Handlers.

[View All Answers](#)

Question - 14:

What is access data using Content Provider:

Ans:

Start by making sure your Android application has the necessary read access permissions. Then, get access to the ContentResolver object by calling getContentResolver() on the Context object, and retrieving the data by constructing a query using ContentResolver.query().

The ContentResolver.query() method returns a Cursor, so you can retrieve data from each column using Cursor methods.



[View All Answers](#)

Question - 15:

What is activity Context?

Ans:

This context is available in an activity. This context is tied to the lifecycle of an activity. The activity context should be used when you are passing the context in the scope of an activity or you need the context whose lifecycle is attached to the current context.

[View All Answers](#)

Question - 16:

Explain me the difference between onCreate() and onStart()?

Ans:

The onCreate() method is called once during the Activity lifecycle, either when the application starts, or when the Activity has been destroyed and then recreated, for example during a configuration change.

The onStart() method is called whenever the Activity becomes visible to the user, typically after onCreate() or onRestart().

[View All Answers](#)

Question - 17:

Explain me difference between Service & Intent Service?

Ans:

* Service is the base class for Android services that can be extended to create any service. A class that directly extends Service runs on the main thread so it will block the UI (if there is one) and should therefore either be used only for short tasks or should make use of other threads for longer tasks.

* IntentService is a subclass of Service that handles asynchronous requests (expressed as "Intents") on demand. Clients send requests through startService(Intent) calls. The service is started as needed, handles each Intent in turn using a worker thread, and stops itself when it runs out of work.

[View All Answers](#)

Question - 18:

Explain me what is a ContentProvider and what is it typically used for?

Ans:

A ContentProvider manages access to a structured set of data. It encapsulates the data and provide mechanisms for defining data security. ContentProvider is the standard interface that connects data in one process with code running in another process.

[View All Answers](#)

Question - 19:

Do you know the languages supported for Android development?

Ans:

Java is the widely used language for Android development.

It also supports C/C++ and when used with Android SDK, it improves the performance speed too.

[View All Answers](#)

Question - 20:

Tell us what is a BuildType in Gradle? And what can you use it for?

Ans:

Build types define properties that Gradle uses when building and packaging your Android app.

* A build type defines how a module is built, for example whether ProGuard is run.

* A product flavour defines what is built, such as which resources are included in the build.

* Gradle creates a build variant for every possible combination of your project's product flavours and build types.

[View All Answers](#)

Question - 21:

Tell us why would you do the setContentView() in onCreate() of Activity class?

Ans:

As onCreate() of an Activity is called only once, this is the point where most initialisation should go. It is inefficient to set the content in onResume() or onStart() (which are called multiple times) as the setContentView() is a heavy operation.

[View All Answers](#)

Question - 22:

Please explain what is a Sticky Intent?

Ans:

Sticky Intents allows communication between a function and a service. sendStickyBroadcast() performs a sendBroadcast(Intent) known as sticky, i.e. the Intent you are sending stays around after the broadcast is complete, so that others can quickly retrieve that data through the return value of registerReceiver(BroadcastReceiver, IntentFilter). For example, if you take an intent for ACTION_BATTERY_CHANGED to get battery change events: When you call registerReceiver() for that action - even with a null BroadcastReceiver - you get the Intent that was last Broadcast for that action. Hence, you can use this to find the state of the battery without necessarily registering for all future state changes in the battery.



[View All Answers](#)

Question - 23:

Do you know what is an Action?

Ans:

Description of the intent. For instance, ACTION_CALL - used to perform calls

[View All Answers](#)

Question - 24:

Please explain the onTrimMemory() method?

Ans:

onTrimMemory(): Called when the operating system has determined that it is a good time for a process to trim unneeded memory from its process. This will happen for example when it goes in the background and there is not enough memory to keep as many background processes running as desired.

Android can reclaim memory for from your app in several ways or kill your app entirely if necessary to free up memory for critical tasks. To help balance the system memory and avoid the system's need to kill your app process, you can implement the ComponentCallbacks2 interface in your Activity classes. The provided onTrimMemory() callback method allows your app to listen for memory related events when your app is in either the foreground or the background, and then release objects in response to app lifecycle or system events that indicate the system needs to reclaim memory.

[View All Answers](#)

Question - 25:

Explain me onSaveInstanceState() and onRestoreInstanceState() in activity?

Ans:

onRestoreInstanceState() - When activity is recreated after it was previously destroyed, we can recover the saved state from the Bundle that the system passes to the activity. Both the onCreate() and onRestoreInstanceState() callback methods receive the same Bundle that contains the instance state information. But because the onCreate() method is called whether the system is creating a new instance of your activity or recreating a previous one, you must check whether the state Bundle is null before you attempt to read it. If it is null, then the system is creating a new instance of the activity, instead of restoring a previous one that was destroyed.

onSaveInstanceState() - is a method used to store data before pausing the activity.

[View All Answers](#)

Question - 26:

Explain me what is the difference between Service and IntentService? How is each used?

Ans:

Service is the base class for Android services that can be extended to create any service. A class that directly extends Service runs on the main thread so it will block the UI (if there is one) and should therefore either be used only for short tasks or should make use of other threads for longer tasks.

IntentService is a subclass of Service that handles asynchronous requests (expressed as "Intents") on demand. Clients send requests through startService(Intent) calls. The service is started as needed, handles each Intent in turn using a worker thread, and stops itself when it runs out of work. Writing an IntentService can be quite simple; just extend the IntentService class and override the onHandleIntent(Intent intent) method where you can manage all incoming requests.

[View All Answers](#)

Question - 27:

Explain me activities?

Ans:

Activities are basically containers or windows to the user interface.

[View All Answers](#)

Question - 28:

Please explain how to prevent the data from reloading and resetting when the screen is rotated?

Ans:

* The most basic approach would be to use a combination of ViewModels and onSaveInstanceState() . So how we do we that?

* Basics of ViewModel: A ViewModel is Lifecycle-Aware. In other words, a ViewModel will not be destroyed if its owner is destroyed for a configuration change (e.g. rotation). The new instance of the owner will just re-connected to the existing ViewModel. So if you rotate an Activity three times, you have just created three different Activity instances, but you only have one ViewModel.

* So the common practice is to store data in the ViewModel class (since it persists data during configuration changes) and use onSaveInstanceState to store small amounts of UI data.

* For instance, let's say we have a search screen and the user has entered a query in the EditText. This results in a list of items being displayed in the RecyclerView. Now if the screen is rotated, the ideal way to prevent resetting of data would be to store the list of search items in the ViewModel and the query text user has entered in the onSaveInstanceState method of the activity.

[View All Answers](#)

Question - 29:

Please explain what is an Intent? Can it be used to provide data to a ContentProvider? Why or why not?

Ans:

The Intent object is a common mechanism for starting new activity and transferring data from one activity to another. However, you cannot start a ContentProvider using an Intent.

When you want to access data in a ContentProvider, you must instead use the ContentResolver object in your application's Context to communicate with the provider as a client. The ContentResolver object communicates with the provider object, an instance of a class that implements ContentProvider. The provider object receives



data requests from clients, performs the requested action, and returns the results.

[View All Answers](#)

Question - 30:

Tell us what is the difference between Mobile Application Testing and Mobile Testing?

Ans:

Mobile app testing is the testing of applications on a device which mainly focuses on functions and features of the application.

And Mobile Testing is the testing of the actual mobile device and focuses on the mobile features like Call, SMS, Contacts, Media Player, inbuilt browsers etc.

[View All Answers](#)

Question - 31:

What is lifecycle of an Activity?

Ans:

* onCreate(): This is when the view is first created. This is normally where we create views, get data from bundles etc.

* onStart(): Called when the activity is becoming visible to the user. Followed by onResume() if the activity comes to the foreground, or onStop() if it becomes hidden.

* onResume(): Called when the activity will start interacting with the user. At this point your activity is at the top of the activity stack, with user input going to it.

* onPause(): Called as part of the activity lifecycle when an activity is going into the background, but has not (yet) been killed.

* onStop(): Called when you are no longer visible to the user.

* onDestroy(): Called when the activity is finishing

* onRestart(): Called after your activity has been stopped, prior to it being started again

[View All Answers](#)

Question - 32:

Please explain the difference between FLAG_ACTIVITY_CLEAR_TASK and FLAG_ACTIVITY_CLEAR_TOP?

Ans:

* FLAG_ACTIVITY_CLEAR_TASK is used to clear all the activities from the task including any existing instances of the class invoked. The Activity launched by intent becomes the new root of the otherwise empty task list. This flag has to be used in conjunction with FLAG_ACTIVITY_NEW_TASK.

* FLAG_ACTIVITY_CLEAR_TOP on the other hand, if set and if an old instance of this Activity exists in the task list then barring that all the other activities are removed and that old activity becomes the root of the task list. Else if there's no instance of that activity then a new instance of it is made the root of the task list. Using FLAG_ACTIVITY_NEW_TASK in conjunction is a good practice, though not necessary.

[View All Answers](#)

Question - 33:

Tell us what is the relationship between the life cycle of an AsyncTask and an Activity? What problems can this result in? How can these problems be avoided?

Ans:

An AsyncTask is not tied to the life cycle of the Activity that contains it. So, for example, if you start an AsyncTask inside an Activity and the user rotates the device, the Activity will be destroyed (and a new Activity instance will be created) but the AsyncTask will not die but instead goes on living until it completes.

Then, when the AsyncTask does complete, rather than updating the UI of the new Activity, it updates the former instance of the Activity (i.e., the one in which it was created but that is not displayed anymore!). This can lead to an Exception (of the type java.lang.IllegalArgumentException: View not attached to window manager if you use, for instance, findViewById to retrieve a view inside the Activity).

There's also the potential for this to result in a memory leak since the AsyncTask maintains a reference to the Activity, which prevents the Activity from being garbage collected as long as the AsyncTask remains alive.

For these reasons, using AsyncTasks for long-running background tasks is generally a bad idea. Rather, for long-running background tasks, a different mechanism (such as a service) should be employed.

[View All Answers](#)

Question - 34:

What is fragment lifecycle?

Ans:

* onAttach() : The fragment instance is associated with an activity instance. The fragment and the activity is not fully initialized. Typically you get in this method a reference to the activity which uses the fragment for further initialization work.

* onCreate() : The system calls this method when creating the fragment. You should initialize essential components of the fragment that you want to retain when the fragment is paused or stopped, then resumed.

* onCreateView() : The system calls this callback when it's time for the fragment to draw its user interface for the first time. To draw a UI for your fragment, you must return a View component from this method that is the root of your fragment's layout. You can return null if the fragment does not provide a UI.

* onActivityCreated() : The onActivityCreated() is called after the onCreateView() method when the host activity is created. Activity and fragment instance have been created as well as the view hierarchy of the activity. At this point, view can be accessed with the findViewById() method. example. In this method you can instantiate objects which require a Context object

* onStart() : The onStart() method is called once the fragment gets visible.

* onResume() : Fragment becomes active.

* onPause() : The system calls this method as the first indication that the user is leaving the fragment. This is usually where you should commit any changes that should be persisted beyond the current user session.

* onStop() : Fragment going to be stopped by calling onStop()

* onDestroyView() : Fragment view will destroy after call this method

* onDestroy() : called to do final clean up of the fragment's state but Not guaranteed to be called by the Android platform.

[View All Answers](#)

Question - 35:

Tell us what is the difference between a fragment and an activity? Explain the relationship between the two?



Ans:

An activity is typically a single, focused operation that a user can perform (such as dial a number, take a picture, send an email, view a map, etc.). Yet at the same time, there is nothing that precludes a developer from creating an activity that is arbitrarily complex.

Activity implementations can optionally make use of the Fragment class for purposes such as producing more modular code, building more sophisticated user interfaces for larger screens, helping scale applications between small and large screens, and so on. Multiple fragments can be combined within a single activity and, conversely, the same fragment can often be reused across multiple activities. This structure is largely intended to foster code reuse and facilitate economies of scale.

A fragment is essentially a modular section of an activity, with its own lifecycle and input events, and which can be added or removed at will. It is important to remember, though, that a fragment's lifecycle is directly affected by its host activity's lifecycle; i.e., when the activity is paused, so are all fragments in it, and when the activity is destroyed, so are all of its fragments.

[View All Answers](#)

Question - 36:

Tell us how do you supply construction arguments into a Fragment?

Ans:

Construction arguments for a Fragment are passed via Bundle using the `Fragment#setArgument(Bundle)` method. The passed-in Bundle can then be retrieved through the `Fragment#getArguments()` method in the appropriate Fragment lifecycle method.

It is a common mistake to pass in data through a custom constructor. Non-default constructors on a Fragment are not advisable because the Fragment may be destroyed and recreated due to a configuration change (e.g. orientation change). Using `#setArguments()/getArguments()` ensures that when the Fragment needs to be recreated, the Bundle will be appropriately serialized/deserialized so that construction data is restored.

[View All Answers](#)

Question - 37:

Tell us why bytecode cannot be run in Android?

Ans:

Android uses DVM (Dalvik Virtual Machine) rather using JVM (Java Virtual Machine).

[View All Answers](#)

Question - 38:

Do you know launch modes in Android?

Ans:

- * Standard: It creates a new instance of an activity in the task from which it was started. Multiple instances of the activity can be created and multiple instances can be added to the same or different tasks.

- * Eg: Suppose there is an activity stack of A -> B -> C.

- * Now if we launch B again with the launch mode as "standard", the new stack will be A -> B -> C -> B.

- * SingleTop: It is the same as the standard, except if there is a previous instance of the activity that exists in the top of the stack, then it will not create a new instance but rather send the intent to the existing instance of the activity.

- * Eg: Suppose there is an activity stack of A -> B.

- * Now if we launch C with the launch mode as "singleTop", the new stack will be A -> B -> C as usual.

- * Now if there is an activity stack of A -> B -> C.

- * If we launch C again with the launch mode as "singleTop", the new stack will still be A -> B -> C.

- * SingleTask: A new task will always be created and a new instance will be pushed to the task as the root one. So if the activity is already in the task, the intent will be redirected to `onNewIntent()` else a new instance will be created. At a time only one instance of activity will exist.

- * Eg: Suppose there is an activity stack of A -> B -> C -> D.

- * Now if we launch D with the launch mode as "singleTask", the new stack will be A -> B -> C -> D as usual.

- * Now if there is an activity stack of A -> B -> C -> D.

- * If we launch activity B again with the launch mode as "singleTask", the new activity stack will be A -> B. Activities C and D will be destroyed.

- * SingleInstance: Same as single task but the system does not launch any activities in the same task as this activity. If new activities are launched, they are done so in a separate task.

- * Eg: Suppose there is an activity stack of A -> B -> C -> D. If we launch activity B again with the launch mode as "singleInstance", the new activity stack will be:

- * Task1 - A -> B -> C

- * Task2 - D

[View All Answers](#)

Question - 39:

Tell me two ways to clear the back stack of Activities when a new Activity is called using intent?

Ans:

The first approach is to use a `FLAG_ACTIVITY_CLEAR_TOP` flag. The second way is by using `FLAG_ACTIVITY_CLEAR_TASK` and `FLAG_ACTIVITY_NEW_TASK` in conjunction.

[View All Answers](#)

Question - 40:

Tell us difference between Service, Intent Service, AsyncTask & Threads?

Ans:

- * Android service is a component that is used to perform operations on the background such as playing music. It doesn't have any UI (user interface). The service runs in the background indefinitely even if application is destroyed.

- * AsyncTask allows you to perform asynchronous work on your user interface. It performs the blocking operations in a worker thread and then publishes the results on the UI thread, without requiring you to handle threads and/or handlers yourself.

- * IntentService is a base class for Services that handle asynchronous requests (expressed as Intents) on demand. Clients send requests through `startService(Intent)` calls; the service is started as needed, handles each Intent in turn using a worker thread, and stops itself when it runs out of work.

- * A thread is a single sequential flow of control within a program. Threads can be thought of as mini-processes running within a main process.



[View All Answers](#)

Question - 41:

What is difference between Serializable and Parcelable?

Ans:

Serialization is the process of converting an object into a stream of bytes in order to store an object into memory, so that it can be recreated at a later time, while still keeping the object's original state and data.

[View All Answers](#)

Question - 42:

Please explain what are intent Filters?

Ans:

Specifies the type of intent that the activity/service can respond to.

[View All Answers](#)

Question - 43:

Explain me three common use cases for using an Intent?

Ans:

Common use cases for using an Intent include:

- * To start an activity: You can start a new instance of an Activity by passing an Intent to startActivity() method.
- * To start a service: You can start a service to perform a one-time operation (such as download a file) by passing an Intent to startService().
- * To deliver a broadcast: You can deliver a broadcast to other apps by passing an Intent to sendBroadcast(), sendOrderedBroadcast(), or sendStickyBroadcast().

[View All Answers](#)

Question - 44:

Please explain what is difference between Serializable and Parcelable ? Which is best approach in Android?

Ans:

Serializable is a standard Java interface. You simply mark a class Serializable by implementing the interface, and Java will automatically serialize it in certain situations.

Parcelable is an Android specific interface where you implement the serialization yourself. It was created to be far more efficient than Serializable, and to get around some problems with the default Java serialization scheme.

[View All Answers](#)

Question - 45:

Explain me what is a broadcast receiver?

Ans:

The broadcast receiver communicates with the operation system messages such as "check whether an internet connection is available," what the battery level should be, etc.

[View All Answers](#)

Question - 46:

Explain me what are the advantages of Android Operating System?

Ans:

It is an open-source and platform independent. It supports various technologies like Bluetooth, Wi-Fi, etc

[View All Answers](#)

Question - 47:

Do you know what is Application?

Ans:

The Application class in Android is the base class within an Android app that contains all other components such as activities and services. The Application class, or any subclass of the Application class, is instantiated before any other class when the process for your application/package is created.

[View All Answers](#)

Question - 48:

What are fragments?

Ans:

Fragment is a UI entity attached to Activity. Fragments can be reused by attaching in different activities. Activity can have multiple fragments attached to it. Fragment must be attached to an activity and its lifecycle will depend on its host activity.

[View All Answers](#)

Question - 49:

Explain me what is an intent?

**Ans:**

Intents are messages that can be used to pass information to the various components of android. For instance, launch an activity, open a webview etc. Two types of intents-

- * Implicit: Implicit intent is when you call system default intent like send email, send SMS, dial number.
- * Explicit: Explicit intent is when you call an application activity from another activity of the same application.

[View All Answers](#)

Question - 50:

Please explain what is a Job Scheduling?

Ans:

Job Scheduling api, as the name suggests, allows to schedule jobs while letting the system optimize based on memory, power, and connectivity conditions. The JobScheduler supports batch scheduling of jobs. The Android system can combine jobs so that battery consumption is reduced. JobManager makes handling uploads easier as it handles automatically the unreliability of the network. It also survives application restarts. Some scenarios:

- * Tasks that should be done once the device is connect to a power supply
- * Tasks that require network access or a Wi-Fi connection.
- * Task that are not critical or user facing
- * Tasks that should be running on a regular basis as batch where the timing is not critical
- * You can click on this link to learn more about Job Schedulers.

[View All Answers](#)

Question - 51:

Explain me difference between AsyncTasks & Threads?

Ans:

- * Thread should be used to separate long running operations from main thread so that performance is improved. But it can't be cancelled elegantly and it can't handle configuration changes of Android. You can't update UI from Thread.
- * AsyncTask can be used to handle work items shorter than 5ms in duration. With AsyncTask, you can update UI unlike java Thread. But many long running tasks will choke the performance.

[View All Answers](#)

Question - 52:

What are content providers?

Ans:

A ContentProvider provides data from one application to another, when requested. It manages access to a structured set of data. It provides mechanisms for defining data security. ContentProvider is the standard interface that connects data in one process with code running in another process.

When you want to access data in a ContentProvider, you must instead use the ContentResolver object in your application's Context to communicate with the provider as a client. The provider object receives data requests from clients, performs the requested action, and returns the results.

[View All Answers](#)

Question - 53:

Explain me how does the activity respond when the user rotates the screen?

Ans:

When the screen is rotated, the current instance of activity is destroyed a new instance of the Activity is created in the new orientation. The onRestart() method is invoked first when a screen is rotated. The other lifecycle methods get invoked in the similar flow as they were when the activity was first created.

[View All Answers](#)

Question - 54:

Explain me scenario in which only onDestroy is called for an activity without onPause() and onStop()?

Ans:

If finish() is called in the onCreate method of an activity, the system will invoke onDestroy() method directly.

[View All Answers](#)

Question - 55:

Explain me what is the Android Application Architecture?

Ans:

Android application architecture has the following components:

- * Services - It will perform background functionalities
- * Intent - It will perform the inter connection between activities and the data passing mechanism
- * Resource Externalization - strings and graphics
- * Notification - light, sound, icon, notification, dialog box and toast
- * Content Providers - It will share the data between applications

[View All Answers](#)

Question - 56:

Explain me what is Context?

Ans:

A Context is a handle to the system; it provides services like resolving resources, obtaining access to databases and preferences, and so on. An Android app has



activities. Context is like a handle to the environment your application is currently running in.

[View All Answers](#)

Interview Questions Answers.ORG

Mobile Technologies Most Popular & Related Interview Guides

- 1 : [Android Interview Questions and Answers.](#)
- 2 : [Wireless Communication Interview Questions and Answers.](#)
- 3 : [3G Interview Questions and Answers.](#)
- 4 : [iPhone Interview Questions and Answers.](#)
- 5 : [Mobile App Architect Interview Questions and Answers.](#)
- 6 : [Blackberry Interview Questions and Answers.](#)
- 7 : [App Store Optimizer \(ASO\) Interview Questions and Answers.](#)
- 8 : [Mobile Application Interview Questions and Answers.](#)
- 9 : [Android Software Engineer Interview Questions and Answers.](#)
- 10 : [Mobile Application Developer Interview Questions and Answers.](#)

Follow us on FaceBook

www.facebook.com/InterviewQuestionsAnswers.Org

Follow us on Twitter

<https://twitter.com/InterviewQA>

For any inquiry please do not hesitate to contact us.

Interview Questions Answers.ORG Team

[https://InterviewQuestionsAnswers.ORG/
support@InterviewQuestionsAnswers.ORG](https://InterviewQuestionsAnswers.ORG/support@InterviewQuestionsAnswers.ORG)