How Android code is compiled?

The compilers convert your source code into DEX (Dalvik Executable) files, which include the bytecode that runs on Android devices, and everything else into compiled resources. The packager combines the DEX files and compiled resources into an APK or AAB, depending on the chosen build target.

What are the benefits of intents in Android?

Android Intent is the message that is passed between components such as activities, content providers, broadcast receivers, services etc.

Android uses Intent for communicating between the components of an Application and also from one application to another application.

Intent are the objects which is used in android for passing the information among Activities in an Application and from one app to another also. Intent are used for communicating between the Application components and it also provides the connectivity between two apps.

For example: Intent facilitate you to redirect your activity to another activity on occurrence of any event. By calling, startActivity() you can perform this task.

```
Intent intent = new Intent(getApplicationContext(), SecondActivity.class);

startActivity(intent);
```

In the above example, foreground activity is getting redirected to another activity i.e. SecondActivity.java. getApplicationContext() returns the context for your foreground activity.

How to differentiate implicit and explicit intents in Android?

Difference Table

Explicit Intent	Implicit Intent
As already mentioned above explicit intents are those in which the user has a clear vision and knows exactly which activity can handle the requests. **Example**: When you have a Listview screen on tap of each item you will go to detail activity **Intent = Intent(applicationContext,DetailActivity::class.java)** **startActivity(intent)**	Implicit intents do not name a specific component like explicit intent, instead declare general action to perform, which allows a component from another app to handle. **Example:** When you tap the share button in any app you can see the Gmail, Bluetooth, and other sharing app options. Here user sends a request is the implicit intent request which can

	be handled by these Gmail, Bluetooth-like app.
Explicit intent can do the specific application action which is set by the code like changing activity, downloading the file in the background, etc.	It specifies the only action to be performed and does not directly specify Android Components.

In explicit intent, you can pass data to other activities by using the putExtra method and retrieve by using getIntent().
 	
Example:
 	
val intent = Intent(this, SecondActivity:: class.java).apply{
 	
putExtra("key","New Value")
 	
}
 	
startActivity(intent)
 	
Second Screen:
 	
val secondIntent = intent.getStringExtra("key")
	Here we just mention the action in the intent and OS decides which applications are suitable to handle the task, action across two different applications.

Explicit intents are used for communication inside the application. Like changing activities inside the application.	They are used for communication across two different applications.
In explicit intent, the action target is delivered even the filter is not consulted.	When you make an implicit call with the intent. OS look at the action and then it matches with all the filters intent-filters of all the registered activities of all application using PackageManager and then populates the result as a list, this process is called as intent Resolution.

What are the steps involved in the execution of an Android program?

See the detail answer on

https://www.geeksforgeeks.org/how-does-android-app-work/

Why activity life cycle is important?

The activity lifecycle is **the set of states an activity can be in during its entire lifetime, from the time it's created to when it's destroyed and the system reclaims its resources**. As the user interacts with your app and other apps on the device, activities move into different states.

What is the difference between onCreate () and onStart ()?

`***onCreate()***`

Called when the activity is first created. This is where you should do all of your normal static set up: create views, bind data to lists, etc. This method also provides you with a Bundle containing the activity's previously frozen state, if there was one. Always followed by onStart().

`***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.

Why do we need to call setContentView () in onCreate () of activity class?

As `onCreate()` of an Activity is called only once, this is the point where most initialization should go: calling `setContentView(int)` to inflate the activity's UI, using findViewById to programmatically interact with widgets in the UI, calling `managedQuery(android.net.Uri, String[], String, String[], String)` to retrieve cursors for data being displayed, etc.

It is inefficient to set the content in `onResume()` or `onStart()` (which are called multiple times) as the `setContentView()` is a heavy operation.

What is the lifecycle of services in Android?

See the detail answer on

https://data-flair.training/blogs/android-service-tutorial/

What are the usage of Android services?

Android service is a component that is used **to perform operations on the background such as playing music, handle network transactions, interacting content providers etc**. It doesn't has any UI (user interface). The service runs in the background indefinitely even if application is destroyed.

Can we start a service without activity? Explain

It's not possible to have a Service on its own as a stand-alone "app". **It needs to be started manually by a user through an Activity** .

What is binding and querying Android service explain?

A bound service is **an implementation of the Service class that allows other applications to bind to it and interact with it**. To provide binding for a service, you must implement the onBind() callback method.

It allows components (such as activities) to bind to the service, send requests, receive responses, and perform interprocess communication (IPC). A bound service typically lives only while it serves another application component and does not run in the background indefinitely.

How do you send a toast message on Android?

Instantiate a Toast object

Use the makeText() method, which takes the following parameters: The application Context . The text that should appear to the user. The duration that the toast should remain on the screen.

Why is it called toast Android?

It's called a toast **because like a piece of bread in a toaster, it momentarily pops up so that the user can see it**. It gets displayed for a certain amount of time and then disappears on its own without any interaction from the user.

What is spinner explain with example?

See the detail answer on

https://www.geeksforgeeks.org/spinner-in-android-using-java-with-example/

How do I send data from BroadcastReceiver to activity?

See the detail answer on

https://www.digitalocean.com/community/tutorials/android-broadcastreceiver-example-tutorial

Where do I register and unregister BroadCast receiver?

If you *register* a *receiver* in onResume() , you should *unregister* it in onPause() to prevent *registering* it multiple times

How is content provider used in Android?

Content providers can **help an application manage access to data stored by itself, stored by other apps, and provide a way to share data with other apps**. They encapsulate the data, and provide mechanisms for defining data security.

How does SQLite work in Android?

SQLite Database is an open-source database provided in Android which is **used to store data inside the user's device in the form of a Text file**. We can perform so many operations on this data such as adding new data, updating, reading, and deleting this data.

Also see this link

https://www.javatpoint.com/android-sqlite-tutorial

What are the 5 types of Android layouts?

See the detail answer on

https://www.tutorialspoint.com/android/android_user_interface_layouts.htm

Difference Between a Fragment and an Activity in Android

Activity is an application component that gives a user interface where the user can interact. **The fragment is only part of an activity, it basically contributes its UI to that activity**. Fragment is dependent on activity. It can't exist independently.