1. What is Application class?
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.

2. What is Toast in Android?

Android Toast can be used to display information for the short period of time. A toast contains message to be displayed quickly and disappears after sometime.

3. Difference between margin & padding?

Padding will be space added inside the container, for instance, if it is a button, padding will be added inside the button. Margin will be space added outside the container.

4. What is the AndroidManifest.xml?

This file is essential in every application. It is declared in the root directory and contains information about the application that the Android system must know before the codes can be executed.

5. What is the difference between a regular .png and a nine-patch image?

It is one of a resizable bitmap resource which is being used as backgrounds or other images on the device. The NinePatch class allows drawing a bitmap in nine sections. The four corners are unscaled; the middle of the image is scaled in both axes, the four edges are scaled into one axis.

6. What is Adapters?

An adapter responsible for converting each data entry into a View that can then be added to the AdapterView (ListView/RecyclerView).

7. What is a singleton class in Android?

A singleton class is a class which can create only an object that can be shared all other classes.

8. What is an intent?

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.*

9. What is a BuildType in Gradle? And what can you use it for?

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

1. *A build type defines how a module is built, for example whether ProGuard is run.*
2. *A product flavour defines what is built, such as which resources are included in the build.*
3. *Gradle creates a build variant for every possible combination of your project's product flavours and build types.*

10. Scenario in which only onDestroy is called for an activity without onPause() and onStop()?

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

11. What is the Android Application Architecture?

Android application architecture has the following components:

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

12. Android Architecture Components?

A collection of libraries that help you design robust, testable, and maintainable apps.

- *Room*
- *Live Data*
- *ViewModel*
- *Data Binding*
- *Lifecycles*

13. Lifecycle of an Activity

- `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*

14. Describe fragment lifecycle

- `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.*

15. When should you use a fragment rather than an activity?

- *When there are ui components that are going to be used across multiple activities.*
- *When there are multiple views that can be displayed side by side (viewPager tabs)*
- *When you have data that needs to be persisted across Activity restarts (such as retained fragments)*

16. onSavedInstanceState() and onRestoreInstanceState() in activity?

`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.*

17. Launch modes in Android?

- ***Standard***
- ***SingleTop***
- ***SingleTask***
- ***SingleInstance***

18. How does the activity respond when the user rotates the screen?

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.

19. What is a broadcast receiver?

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

20. What's the difference between FLAG_ACTIVITY_CLEAR_TASK and FLAG_ACTIVITY_CLEAR_TOP?

***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.*