

Android App Lifecycle

Your Name:

1. What is the primary reason for understanding the Android app lifecycle?

- ☐ A. To write more efficient Java code.
 - ☐ B. To avoid using the Toast class.
 - ☐ C. To simplify the structure of Android projects.
 - ☐ D. To know where to place code to handle specific app events.
-

2. What does method overriding in the context of the Android lifecycle allow developers to do?

- ☐ A. Replace the entire Android API with custom code.
 - ☐ B. Prevent the Android OS from calling certain methods.
 - ☐ C. Add specific functionality at predefined points in the app's lifecycle.
 - ☐ D. Simplify the process of debugging Android applications.
-

3. Why is it important for an app to handle the `onPause` method correctly?

- ☐ A. To save any unsaved user data before the app is interrupted.
 - ☐ B. To guarantee the app restarts from the beginning.
 - ☐ C. To prevent the app from being destroyed by the operating system.
 - ☐ D. To ensure the app uses minimal processing power.
-

4. What is the purpose of calling

`super.onCreate(savedInstanceState)` within the `onCreate` method?

- ☐ A. To execute code in a superclass before any other actions.
 - ☐ B. To ensure that the Android system can properly initialize the activity.
 - ☐ C. To load previously saved user preferences.
 - ☐ D. To skip the official Android version of onCreate.
-

5. In the Android lifecycle, what is the difference between the `onStop` and `onDestroy` methods?

- ☐ A. `onStop` is called when the activity is no longer visible, while `onDestroy` is called before the activity is killed.
 - ☐ B. `onStop` is called before `onCreate`, while `onDestroy` is called after `onResume`.
 - ☐ C. `onStop` is called when the activity is killed, while `onDestroy` is called when the activity becomes visible.
 - ☐ D. `onStop` and `onDestroy` are interchangeable and perform the same function.
-

6. An Android app is running in the foreground. A user switches to another app. Which lifecycle method is called on the original app before the switch?

- ☐ A. onStart
 - ☐ B. onResume
 - ☐ C. onDestroy
 - ☐ D. onPause
-

7. What does the `@Override` annotation signify in Android development?

- ☐ A. The method reimplements a method inherited from a parent class.
 - ☐ B. The method is part of the core Android operating system.
 - ☐ C. The method can only be called by the system, not by the developer.
 - ☐ D. The method is deprecated and should not be used.
-

8. If an Android app needs to release system resources, such as database connections or network sockets, during which lifecycle phase should it do so?

- ☐ A. `onStop`
 - ☐ B. `onPause`
 - ☐ C. `onResume`
 - ☐ D. `onStart`
-

9. What is the significance of the `savedInstanceState` parameter in the `onCreate` method?

- ☐ A. It allows you to pass data between activities.
 - ☐ B. It provides access to the application's theme.
 - ☐ C. It holds a reference to the layout file.
 - ☐ D. It saves the state of the activity so it can be recreated.
-

10. An app is minimized, and the system is running low on resources. Which method is the last guarantee that will be called before the app is potentially killed to free up memory?

- ☐ A. `onStart`
 - ☐ B. `onPause`
 - ☐ C. `onStop`
 - ☐ D. `onDestroy`
-

11. Which lifecycle methods are typically called when an activity first starts?

- ☐ A. onCreate, onStart, onResume
 - ☐ B. onPause, onStop, onDestroy
 - ☐ C. onCreate, onStop, onDestroy
 - ☐ D. onStart, onResume, onPause
-

12. What is the purpose of the `Toast` class in Android development, as mentioned in the text?

- ☐ A. To store user preferences.
 - ☐ B. To handle network connections.
 - ☐ C. To manage background processes.
 - ☐ D. To send pop-up messages to the user.
-

13. Why does Android call methods contained within the Activity class even if they are not directly visible in the Java code written by the developer?

- ☐ A. To ensure code security and prevent unauthorized access.
 - ☐ B. To facilitate communication between different apps.
 - ☐ C. To manage the app's lifecycle events at appropriate times.
 - ☐ D. To optimize system resource usage.
-

14. What should you do in the `onCreateOptionsMenu` method?

- ☐ A. Save user data.
 - ☐ B. Handle menu item selection.
 - ☐ C. Load the menu from the XML file.
 - ☐ D. Release system resources.
-

15. What is the purpose of the `onOptionsItemSelected` method?

- ☐ A. To release resources.
 - ☐ B. To save app state.
 - ☐ C. To create the options menu.
 - ☐ D. To handle clicks on items in the options menu.
-

16. What does extending a class allow you to do?

- ☐ A. Create a new class based on an existing one, inheriting its properties and methods.
 - ☐ B. Use the class without modifying it.
 - ☐ C. Hide a class from other parts of the code.
 - ☐ D. Import a class for use in your code.
-

17. What should an Android app do in its `onDestroy` method?

- ☐ A. Save user data.
 - ☐ B. Display a goodbye message.
 - ☐ C. Release resources and perform final cleanup.
 - ☐ D. Restart the app.
-

18. What does the `super.on...()` syntax inside lifecycle methods accomplish?

- ☐ A. Prevents the method from being called again.
 - ☐ B. Calls the original method's code in addition to the overridden code.
 - ☐ C. Replaces the original method entirely.
 - ☐ D. Skips the original method's execution.
-

19. An app is running and receives a phone call. What lifecycle method is called?

- ☐ A. onDestroy
 - ☐ B. onPause
 - ☐ C. onStart
 - ☐ D. onResume
-

20. In which lifecycle method should you load user's data?

- ☐ A. onCreate
 - ☐ B. onResume
 - ☐ C. onStop
 - ☐ D. onPause
-

21. If an activity is stopped, but not destroyed, what lifecycle method is called when the activity is revisited?

- ☐ A. onRestart()
 - ☐ B. onStop()
 - ☐ C. onStart()
 - ☐ D. onCreate()
-

22. Which method is called immediately before the activity becomes visible to the user?

- ☐ A. onResume()
 - ☐ B. onStart()
 - ☐ C. onPause()
 - ☐ D. onCreate()
-

23. Which lifecycle method is called when the activity is no longer visible to the user?

- ☐ A. onStop()
 - ☐ B. onRestart()
 - ☐ C. onPause()
 - ☐ D. onDestroy()
-

24. Which lifecycle method is called when the activity is being finished or destroyed by the system?

- ☐ A. onPause()
 - ☐ B. onDestroy()
 - ☐ C. onRestart()
 - ☐ D. onStop()
-

25. What is the main purpose of the Android lifecycle?

- ☐ A. To handle network requests
 - ☐ B. To manage the user interface
 - ☐ C. To respond to the current state of an Activity
 - ☐ D. To store data on the device
-

26. Which configuration change will typically cause an Activity to be destroyed and recreated?

- ☐ A. Changing the screen orientation
 - ☐ B. Switching between applications
 - ☐ C. Changing the volume
 - ☐ D. Receiving a phone call
-

27. What is the proper use for an `Intent` in the context of the Android lifecycle?

- ☐ A. To start a new Activity or Service
 - ☐ B. To manage UI layouts
 - ☐ C. To manage system resources during an Activity's lifecycle
 - ☐ D. To manage the Activity's state persistence
-

28. During which of the following lifecycle phases should you typically unregister broadcast receivers?

- ☐ A. `onResume()`
 - ☐ B. `onCreate()`
 - ☐ C. `onPause()` or `onStop()`
 - ☐ D. `onRestart()`
-

29. When should you call `finish()` on an Activity?

- ☐ A. During screen orientation changes
 - ☐ B. Whenever the Activity is paused
 - ☐ C. At the start of its lifecycle
 - ☐ D. When the Activity completes its task or should be closed
-

30. What is the main purpose of the `onSaveInstanceState()` method?

- ☐ A. To save user preferences
 - ☐ B. To save permanent data
 - ☐ C. To save temporary UI state data before an Activity may be destroyed
 - ☐ D. To save network connection status
-

31. In which scenario would you use `ViewModel` with the lifecycle?

- ☐ A. For storing and managing UI-related data in a lifecycle-conscious way
 - ☐ B. For processing network requests in the background
 - ☐ C. For handling permissions directly in the UI
 - ☐ D. For ensuring tasks are executed only after an Activity is in the `RESUMED` state
-

32. How does `LiveData` work with the lifecycle?

- ☐ A. It helps to manage long-running tasks irrespective of lifecycles
 - ☐ B. It automatically updates or subscribes while only is in the active state
 - ☐ C. It helps to save data after `onDestroy()`
 - ☐ D. It sends notifications to UI components even when they are in the background
-

33. What does the termActivity leak..... refer to?

- ☐ A. Memory leaks
 - ☐ B. Broadcast Receivers
 - ☐ C. UI glitches
 - ☐ D. Security breaches
-

34. How does `Data Binding` relate to the activity lifecycle?

- ☐ A. By allowing UI elements in layouts to bind directly to app data, automatically reflecting changes and updates during the activity lifecycle.
 - ☐ B. By enabling easy saving of activity state between lifecycle events.
 - ☐ C. By automatically updating the dataloaders after onDestroy is called.
 - ☐ D. By allowing background tasks to update the UI directly.
-

35. What advantage do fragments provide an activity through the lifecycle events?

- ☐ A. Enhances process isolation between UI tasks
 - ☐ B. Provides a way to reuse parts of an activity's UI and logic, and help manage UI components more modularly.
 - ☐ C. Increased data security
 - ☐ D. Supports easier integration with background thread.
-

36. In what order must Fragments go through lifecycle events inside of its host Activity?

- ☐ A. The fragment lifecycle is totally independent of the activity's lifecycle.
 - ☐ B. Fragments lifecycle only depends on the host activity's UI events.
 - ☐ C. Detach is always called last, then destroy.
 - ☐ D. Events are somewhat dependent on the Activity's lifecycle, where events like `onAttach` and `onCreate` happen before `onCreateView`.
-

37. What happens when we close the app using the back button?

- ☐ A. onPause, onStop, and onDestroy methods were called
 - ☐ B. onStart, onPause, and onDestroy methods were called
 - ☐ C. onStart, onResume, and onPause methods were called
 - ☐ D. onCreate, onStart, and onResume methods were called
-

38. What methods should be called when start the Lifecycle Demonstration app for the first time?

- ☐ A. the onCreate, onStart, and onResume methods were called
 - ☐ B. the onPause, onStop, and onDestroy methods were called
 - ☐ C. the onCreate, onStop, and onDestroy methods were called
 - ☐ D. the onPause, onStop, and onResume methods were called
-