

1. A Fragment can exist independently without being attached to an Activity.

Answer: False

Explanation: A Fragment must always be associated with an Activity; it cannot exist independently.↳

2. What is the primary purpose of `onSaveInstanceState()` in an Activity?

Answer: b. To save the current state of the Activity in case it is destroyed and recreated

Explanation: This method saves state information in a `Bundle` to restore the Activity later.↳

3. Activities are the only components that can host a Fragment.

Answer: False

Explanation: Fragments can also be hosted by other Fragments (nested Fragments).

4. What happens when you call `finish()` in an Activity?

Answer: a. The current Activity is destroyed and removed from the back stack

Explanation: The system calls `onDestroy()` and removes the Activity from the back stack.↳

5. Which method is called when an Activity is becoming visible to the user?

Answer: a. `onStart()`

Explanation: `onStart()` is called when the Activity becomes visible but not yet interactive.↳

6. Match the Fragment lifecycle method with its corresponding event.

- `onCreateView()` : B) Called to create the Fragment's view hierarchy
 - `onDestroy()` : D) Called when the Fragment is being destroyed
 - `onCreate()` : A) Called when the Fragment is created
 - `onPause()` : C) Called when the Fragment is about to go into the background
-

7. Match the layout attribute with its purpose.

- `layout_margin` : A) Sets the outer space of the view
 - `layout_weight` : D) Defines the proportion of space a view should occupy in a `LinearLayout`
 - `layout_gravity` : C) Specifies how a view should be placed within its parent
 - `layout_padding` : B) Sets the inner space within the view's boundaries
-

8. Which layout allows you to position child views relative to each other?

Answer: a. `RelativeLayout`

Explanation: `RelativeLayout` allows positioning child views relative to each other or the parent.↳

9. The `onPause()` method is guaranteed to be called before an Activity is destroyed.

Answer: False

Explanation: The system may skip `onPause()` in cases where the Activity is forcefully terminated.

10. The `onCreateView()` method in a Fragment is used to create and return the Fragment's view hierarchy.

Answer: True

Explanation: `onCreateView()` is responsible for inflating and returning the Fragment's UI.

11. Match the Fragment-related method with its purpose.

- `replace()` : D) Replaces an existing Fragment with a new one
 - `commit()` : C) Commits a Fragment transaction
 - `addToBackStack()` : B) Adds a Fragment transaction to the back stack
 - `findFragmentById()` : A) Locates a Fragment by its ID
-

12. In Android, a layout defines the structure for a user interface (UI).

Answer: True

Explanation: Layouts define the arrangement of views in an Activity or Fragment.

13. A `RelativeLayout` allows you to position its child views in relation to other child views or the parent container.

Answer: True

Explanation: This is the primary purpose of `RelativeLayout`.

14. Match the Activity lifecycle method with the correct sequence.

- `onStop()` : D) The Activity is no longer visible to the user
 - `onStart()` : A) The Activity becomes visible to the user
 - `onPause()` : C) The Activity loses focus but remains visible
 - `onResume()` : B) The Activity comes to the foreground and becomes interactive
-

15. Which of the following methods is used to add a Fragment to an Activity?

Answer: c. `add()`

Explanation: `add()` is used to add a Fragment to an Activity's container.

16. What is the purpose of `FragmentManager` in Android development?

Answer: a. To perform operations such as adding, removing, or replacing fragments

Explanation: `FragmentManager` is used to manage Fragment-related operations.

17. The `onCreate()` method is called only once throughout the lifetime of an Activity.

Answer: True

Explanation: `onCreate()` is called once when the Activity is created and initializes resources.↳

18. Match the Activity state with its description.

- Paused: B) The Activity is visible but not in the foreground
- Stopped: C) The Activity is no longer visible and may be killed by the system
- Destroyed: D) The Activity is terminated and removed from memory
- Active: A) The Activity is running and interacting with the user

19. The `LinearLayout` arranges its children in a single column or row, either horizontally or vertically.

Answer: True

Explanation: The `LinearLayout` arranges child views in a straight line based on its `orientation` attribute.