**Android interview questions**

1. What is android?
2. What is Linux kernel?
3. What is ART?
4. What is android architecture?
5. What is android runtime?
6. How many layers in android architecture?
7. Specify the name of android architecture?a
8. Functionality of each layer of android architecture?
9. What is activity life cycle?
10. Methods of activity life cycle? Order of life cycle+ methods?
11. Which method of ALC call and when?
12. What is log statements?
13. Different types of log statements?
14. How to connect XML design to kotlin/java files?
15. What is bundles?
16. How to send data/images one activity to another?
17. What is the syntax to set background of XML layout form kotlin/java?
18. What is difference between activity and fragment?
19. Fragment life cycle method and order?
20. Is it possible to run fragment without activity?
21. What is syntax to move or replace the fragment to another?
22. What is custom view?
23. How to implement custom views?
24. What is adapter? How to create custom adapter? Why use custom adapter?
25. What is view holder? How to create view holder? Why use it?
26. Difference between list view and spinner?
27. What is listener? Type of listener? Why they used?
28. What is the name of event listener of list view?
29. What is the name of event listener of recycler view?
30. What is the name of event listener of image view?
31. How recycler view work?
32. Internal implementation of recycler view?
33. What is Gradle? Purpose of Gradle?
34. What is SQLite open helper class? And its mandatory methods?
35. What is the query or syntax to insert, delete, update data in SQLite database?
36. What Is content value?
37. What is google android SDK?
38. What are the basic components and four pillar of android architecture?
39. What is ANR?
40. Difference between onMeasure vs onLayout?
41. View lifecycle method invalidate vs requestlayout?
42. What is launch modes?
43. Different types of launch mode? Difference between single Instance vs single top?
44. What is thread handler loopers?