ANDROID DEVELOPMENT LAB VIVA QUESTIONS & ANSWERS

1. Login Form Authentication 🔐

- **What is LinearLayout and how is it used in login forms?**
  - LinearLayout arranges UI elements either horizontally or vertically. In login forms, it's typically used to stack fields like username, password, and button vertically.
- **How do you implement EditText for username and password?**
  - Use `<EditText>` in XML, set `android:inputType="textEmailAddress"` for username and `android:inputType="textPassword"` for password.
- **How to validate credentials in Android?**
  - Use `TextUtils.isEmpty()` to check if fields are empty and validate credentials against stored data (SharedPreferences or database).
- **Explain Toast messages and their implementation**
  - Toast messages are brief messages that appear on the screen. Example: `Toast.makeText(context, message, Toast.LENGTH_SHORT).show();`
- **How to handle button clicks in Android?**
  - Set an `OnClickListener()` on buttons to handle click events: `button.setOnClickListener(view -> { // action });`
- **What are the best practices for form validation?**
  - Validate fields for emptiness, proper formats (e.g., email), and provide clear feedback for errors before submitting data.

---

2. Activity Lifecycle Demo 🔄

- **Explain the complete Android Activity lifecycle**
  - The activity lifecycle includes `onCreate()`, `onStart()`, `onResume()`, `onPause()`, `onStop()`, `onDestroy()`. These methods manage UI updates, background tasks, and resource allocation.
- **What are the different lifecycle methods?**

- `onCreate()`, `onStart()`, `onResume()`, `onPause()`, `onStop()`, `onDestroy()`, `onRestart()`.

- **When is onCreate() called?**
  - `onCreate()` is called when an activity is first created, typically for initializing UI components and resources.
- **What's the difference between onPause() and onStop()?**
  - `onPause()` is called when the activity loses focus but is still partially visible. `onStop()` is when the activity is no longer visible.
- **How do you implement Toast messages for lifecycle events?**
  - Use `Toast.makeText(this, "Event", Toast.LENGTH_SHORT).show()` in lifecycle methods to display messages at various stages.
- **What happens during configuration changes?**
  - Android automatically restarts the activity on configuration changes (like screen orientation) unless explicitly handled in the `AndroidManifest.xml`.

---

3. Calculator Application 🧮

- **How to implement arithmetic operations in Android?**
  - Use standard arithmetic operators (`+`, `-`, `*`, `/`) inside event listeners. Handle division by zero errors properly.
- **What's the best way to handle input validation?**
  - Use `TextUtils.isDigitsOnly()` to ensure input is numeric or handle exceptions like `NumberFormatException`.
- **How to implement clear functionality?**
  - Reset input fields and result display by setting them to empty or default states.
- **Explain error handling in calculator operations**
  - Use try-catch blocks to handle arithmetic errors like division by zero and display user-friendly messages.
- **How to organize UI elements in LinearLayout?**

- Use `LinearLayout` with vertical orientation to neatly stack buttons and text fields.

- **How to handle decimal calculations?**
  - Ensure calculations preserve precision using `BigDecimal` for accurate decimal handling.

---

4. UI Control Validations ✅

- **How to validate email format in Android?**
  - Use `Patterns.EMAIL_ADDRESS.matcher(email).matches()` to validate the email format.
- **Explain password validation techniques**
  - Validate password length (`password.length() >= 6`) and enforce rules like at least one number, letter, and special character.
- **How to implement real-time error feedback?**
  - Use a `TextWatcher` on `EditText` fields to provide instant feedback and error messages using `setError()`.
- **What is TextUtils and its uses?**
  - `TextUtils` is a utility class for text operations like checking if a string is empty (`TextUtils.isEmpty()`).
- **How to handle empty field validation?**
  - Use `TextUtils.isEmpty()` to check if fields are empty and provide appropriate feedback.
- **Best practices for UI validation?**
  - Validate input before submission, provide clear feedback, and ensure user-friendly error messages.

---

5. User Registration System 📝

- **What is SharedPreferences?**
  - `SharedPreferences` is used for storing key-value pairs of primitive data types (e.g., strings, integers) persistently.
- **How to store and retrieve data using SharedPreferences?**

- Store data using `preferences.edit().putString("key", "value").apply()` and retrieve using `preferences.getString("key", defaultValue)`.

- **How to implement navigation between screens?**
  - Use `Intent` to navigate between activities: `startActivity(new Intent(this, TargetActivity.class));`
- **How to display stored user data?**
  - Retrieve stored data from `SharedPreferences` and display it in UI elements like `TextView`.
- **What are the different modes in SharedPreferences?**
  - Modes include `MODE_PRIVATE` (default), `MODE_WORLD_READABLE`, `MODE_WORLD_WRITEABLE`.
- **How to clear SharedPreferences data?**
  - Use `preferences.edit().clear().apply()` to remove all stored data.

---

6. Facebook-Style Layout 🎨

- **What is RelativeLayout?**
  - `RelativeLayout` arranges UI elements relative to each other, allowing for flexible positioning (e.g., top, bottom, left, right).
- **How does RelativeLayout differ from other layouts?**
  - Unlike `LinearLayout` or `ConstraintLayout`, `RelativeLayout` offers more control over element positioning based on relationships.
- **How to position elements relative to each other?**
  - Use attributes like `android:layout_alignParentTop="true"` or `android:layout_toRightOf="@id/element"` for positioning.
- **How to implement fallback navigation?**
  - Use `onBackPressed()` to manage fallback navigation behavior, typically when using `FragmentTransaction`.
- **What are the common RelativeLayout attributes?**
  - `layout_alignParentTop`, `layout_toRightOf`, `layout_centerInParent`.
- **How to handle different screen sizes?**

- Use `dp` for size units and create responsive layouts with `ConstraintLayout` or `RelativeLayout`.

---

7. Image Toggle Application 🖼️

- **What is FrameLayout?**
  - `FrameLayout` stacks its children on top of each other, making it ideal for displaying images or fragments.
- **How to implement image switching?**
  - Use `ImageView` and dynamically change the `src` property with `imageView.setImageResource()`.
- **How to handle click events for images?**
  - Set an `OnClickListener()` on `ImageView` to toggle the image or perform actions when clicked.
- **What are the advantages of FrameLayout?**
  - Efficient for stacking elements on top of each other, such as images or fragments.
- **How to optimize image loading?**
  - Use libraries like Glide or Picasso for asynchronous image loading and caching.
- **How to handle image resources?**
  - Store images in `res/drawable` and use resource IDs to reference them.

---

8. Adapter & Exception Handling 🛡️

- **What is an ArrayAdapter?**
  - `ArrayAdapter` binds a data source (like an array or list) to a `ListView` or other adapter-based UI elements.
- **How to implement GridView?**
  - Define a `GridView` in XML and set an `ArrayAdapter` to it in your activity or fragment.
- **How to handle exceptions in Android?**

    - Use try-catch blocks for common exceptions like `NullPointerException`, `FileNotFoundException`, etc.
- **What are common exceptions in Android?**
    - Common exceptions include `NullPointerException`, `IOException`, `NetworkOnMainThreadException`.
- **How to implement click events in GridView?**
    - Use `setOnItemClickListener()` to handle clicks on GridView items.
- **Best practices for exception handling?**
    - Catch specific exceptions, log errors for debugging, and provide appropriate user feedback for exceptions.

---

9. Multi-Activity Navigation 🚀

- **What are Intents?**
    - `Intents` are used to start activities or services and can carry data between them.
- **Types of Intents and their uses?**
    - **Explicit Intents**: Start a specific activity or service.
    - **Implicit Intents**: Perform an action, like sending an email or viewing a URL.
- **How to pass data between activities?**
    - Use `putExtra()` to pass data with the `Intent` and retrieve it using `getIntent()` in the target activity.
- **How to handle activity lifecycle during navigation?**
    - Activities have their own lifecycle methods. Manage resources properly using methods like `onPause()` and `onStop()`.
- **What is the back stack?**
    - The back stack maintains a stack of activities that users can navigate back through using the back button.
- **How to implement up navigation?**

- Use `NavController` or `onSupportNavigateUp()` for navigation, and define the parent activity in the `AndroidManifest.xml`

10. ListView Implementation 📜

- **What is ListView?**
  - `ListView` is a UI element that displays a scrollable list of items. Each item can be defined by a simple or custom layout.
- **How to implement ArrayAdapter with ListView?**
  - Create an `ArrayAdapter` with your data and set it to the `ListView` using `setAdapter()`. Example:

```
java
Copy code
ArrayAdapter<String> adapter = new ArrayAdapter<>(thi
s, android.R.layout.simple_list_item_1, data);
listView.setAdapter(adapter);
```

- **How to handle ListView item clicks?**
  - Use `setOnItemClickListener()` to listen for item clicks:

```
java
Copy code
listView.setOnItemClickListener((parent, view, positi
on, id) -> {
    // Handle item click
});
```

- **Difference between ListView and RecyclerView?**
  - `RecyclerView` is more efficient and flexible than `ListView`, allowing advanced layouts and view recycling for large datasets.
- **How to customize ListView items?**
  - Create a custom layout for each list item and use a custom `ArrayAdapter` or `BaseAdapter` to bind data to the layout.

- **How to implement efficient scrolling?**
  - Use `RecyclerView` instead of `ListView` for better performance with large datasets, and make sure the adapter properly recycles views.

---

11. Spinner Component 🔄

- **What is a Spinner?**
  - `Spinner` is a UI widget that provides a dropdown menu for the user to select from a list of options.
- **How to populate Spinner using ArrayAdapter?**
  - Create an `ArrayAdapter` and set it on the `Spinner` using `setAdapter()`. Example:

```
java
Copy code
ArrayAdapter<String> adapter = new ArrayAdapter<>(thi
s, android.R.layout.simple_spinner_item, data);
adapter.setDropDownViewResource(android.R.layout.simp
le_spinner_dropdown_item);
spinner.setAdapter(adapter);
```

- **How to handle Spinner item selection?**
  - Use `setOnItemSelectedListener()` to listen for item selections:

```
java
Copy code
spinner.setOnItemSelectedListener(new AdapterView.OnI
temSelectedListener() {
    @Override
    public void onItemSelected(AdapterView<?> parent,
View view, int position, long id) {
        // Handle selection
    }

    @Override
    public void onNothingSelected(AdapterView<?> pare
```

```
nt) {}
});
```

- **How to customize Spinner appearance?**
  - Customize the `Spinner` dropdown by using a custom layout for the `ArrayAdapter` and styling it in XML.
- **Different Spinner modes?**
  - **Drop-down mode**: Displays a dropdown list.
  - **Dialog mode**: Displays the list in a dialog box (use `setSpinnerMode()` to change mode).
- **How to set default selection?**
  - Use `spinner.setSelection(position)` to select an item by its position in the adapter.

---

12. SQLite Database Operations 🗄️

- **What is SQLite in Android?**
  - SQLite is a lightweight, embedded relational database used in Android to store local data in tables.
- **How to implement CRUD operations?**
  - **Create**: Use `insert()` to add records.
  - **Read**: Use `query()` or `rawQuery()` to retrieve data.
  - **Update**: Use `update()` to modify existing records.
  - **Delete**: Use `delete()` to remove records.
- **What is SQLiteOpenHelper?**
  - `SQLiteOpenHelper` is a helper class for managing database creation and upgrades. It simplifies handling database versions.
- **How to create and upgrade database?**
  - Override `onCreate()` to create tables and `onUpgrade()` to handle schema changes when updating the database version.
- **How to handle database errors?**

- Use `try-catch` blocks to handle exceptions like `SQLiteException`, and log errors for debugging.

- **Best practices for database operations?**
  - Always close database connections with `db.close()`, use `SQLiteDatabase.beginTransaction()` for bulk operations, and handle exceptions properly.

GENERAL VIVA QUESTIONS

1. **What is Android and the latest version of Android?**
   - Android is an open-source OS for mobile devices. Latest: **Android 15** 📱
2. **What's Activity in Android?**
   - An Activity is a screen in an app with a user interface. 🖥️
3. **Why is XML used for frontend development in Android?**
   - XML is lightweight and separates design from code. 🔄
4. **What are the components of the Android Application?**
   - Activities, Services, Content Providers, Broadcast Receivers, Intents. ⚙️
5. **What is the Dalvik Virtual Machine?**
   - DVM executes Android apps efficiently. ⚡
6. **What is Toast in Android?**
   - A short message displayed on the screen. 📝
7. **What's Service in Android?**
   - Runs background tasks without a UI. 🛠️
8. **What's Content Provider in Android?**
   - Manages access to shared data. 🔐
9. **What's Gradle and write down its usage in Android?**
   - Gradle automates build processes. 🛠️
10. **What's the Difference Between Intent and Intent filters?**

- Intent is a message; Intent filter describes component capabilities. 🔗

11. **What is the AndroidManifest.xml?**
    - Declares app components, permissions, and metadata. 📜
12. **Explain the Activity Lifecycle in brief.**
    - Stages: `onCreate()`, `onStart()`, `onResume()`, `onPause()`, `onStop()`, `onDestroy()`. 🔄
13. **What is View in Android?**
    - Basic UI building block like TextView, Button. 🖼️
14. **What is Android?**
    - Open-source OS for mobile devices. 📱
15. **Who is the founder of Android?**
    - Andy Rubin. 👤
16. **Explain the Android application Architecture.**
    - Components: Services, Intents, Resource Externalization, Notifications, Content Providers. 🏗️
17. **What are the code names of android?**
    - Examples: Cupcake, Donut, Eclair, KitKat, Marshmallow. 🍩
18. **What are the advantages of Android?**
    - Open-source, platform-independent, supports various technologies. 💡
19. **Does android support other languages than java?**
    - Yes, C/C++ via NDK. 🔌
20. **What are the core building blocks of android?**
    - Activity, View, Intent, Service, Content Provider, Fragment. 🧱
21. **What is activity in Android?**
    - A screen with a user interface. 🖥️
22. **What are the life cycle methods of android activity?**
    - `onCreate()`, `onStart()`, `onResume()`, `onPause()`, `onStop()`, `onRestart()`, `onDestroy()`. 🔄

23. **What is intent?**
    - A message to perform actions like opening activities. 📩
24. **How are view elements identified in the android program?**
    - Using `findViewById()`. 🕵️
25. **Define Android toast.**
    - A short message displayed to the user. 🍞
26. **Give a list of important folders in android.**
    - `AndroidManifest.xml`, `build.xml`, `bin/`, `src/`, `res/`, `assets/`. 🗂️
27. **Explain the use of 'bundle' in android?**
    - Used to pass data between activities. 📦
28. **What is an application resource file?**
    - Files used in building the app, like layouts and strings. 📄
29. **What is the use of LINUX ID in android?**
    - Unique ID for tracking processes. 🆔
30. **Can the bytecode written in java be run on android?**
    - No. 🚫
31. **List the various storages that are provided by Android.**
    - Shared Preferences, Internal Storage, External Storage, SQLite Databases, Network Connection. 💾
32. **How are layouts placed in Android?**
    - As XML files. 📝
33. **Where are layouts placed in Android?**
    - In the `layout` folder. 📂
34. **What is the implicit intent in android?**
    - Invokes system components. 🔄
35. **What is explicit intent in android?**
    - Invokes a specific activity class. 🔘
36. **How to call another activity in android?**

- Use Intent and `startActivity()`. 🔄

37. **What is service in android?**
    - Background component for long tasks. 🛠️
38. **What is the name of the database used in android?**
    - SQLite. 💽
39. **What is AAPT?**
    - Android Asset Packaging Tool. 🔧
40. **What is a content provider?**
    - Shares data between applications. 📤
41. **What is fragment?**
    - Part of an activity for reusable UI. 🔲
42. **What is ADB?**
    - Android Debug Bridge. 🔌
43. **What is NDK?**
    - Native Development Kit for C/C++. 🛠️
44. **What is ANR?**
    - Application Not Responding dialog. ⏸️
45. **What is the Google Android SDK?**
    - Toolset for developing Android apps. 🛠️
46. **What is an APK format?**
    - Android Packaging Key for app files. 📦
47. **Which language does Android support to develop an application?**
    - Java and C/C++. 💻
48. **What is ADT in Android?**
    - Android Development Tool. 🛠️
49. **What is View Group in Android?**
    - Collection of views and child views. 🔲
50. **What is the Adapter in Android?**

- Connects data to UI components. 🔗

51. **What is nine-patch images tool in Android?**
    - Tool to stretch images in nine sections. 🖼️
52. **Which kernel is used in Android?**
    - Customized Linux kernel. 🐧
53. **What is application Widgets in Android?**
    - Miniature application views for embedding. 🏠
54. **Which types of flags are used to run an application on Android?**
    - `FLAG_ACTIVITY_NEW_TASK`, `FLAG_ACTIVITY_CLEAR_TOP`. 🚩
55. **What is a singleton class in Android?**
    - Class with only one shared object. 🔑
56. **What is sleep mode in Android?**
    - CPU sleeps, only essential services run. 💤
57. **What do you mean by a drawable folder in Android?**
    - Folder for visual resources. 🎨
58. **What is DDMS?**
    - Dalvik Debug Monitor Server. 🔧
59. **Define Android Architecture?**
    - Consists of Linux Kernel, Libraries, Android Framework, Applications. 🏗️
60. **What is a portable wi-fi hotspot?**
    - Shares internet connection with other devices. 🌐
61. **Name the dialog box which is supported by Android?**
    - Alert Dialog, Progress Dialog, Date Picker Dialog, Time Picker Dialog. 💬
62. **Name some exceptions in Android?**
    - `InflateException`, `Surface.OutOfResourceException`, `WindowManager.BadTokenException`. ⚠️
63. **What are the basic tools used to develop an Android app?**

- JDK, Eclipse+ADT plugin, SDK Tools. 🛠️

34. **What are the key components of an Android application?**
    - Activities, Services, Broadcast Receivers, Content Providers, Intents. ⚙️
35. **What is an Activity in Android? Can you explain its lifecycle?**
    - Activity is a UI screen. Lifecycle: `onCreate()`, `onStart()`, `onResume()`, `onPause()`, `onStop()`, `onDestroy()`, `onRestart()`. 🔄
36. **What is an Intent in Android? What are the different types of Intents?**
    - Intent requests actions. Types: **Explicit**, **Implicit**. 📩
37. **What is a Fragment, and how is it different from an Activity?**
    - Fragment is part of an activity. It's modular and reusable. 🔲
38. **What are Gradle?**
    - Build automation tool for Android projects. 🔧
39. **What is the AndroidManifest.xml file, and what is its purpose?**
    - Declares app components, permissions, and configurations. 📜
70. **What is a Layout in Android?**
    - Container for arranging UI components. 📐
71. **What are the different types of Layouts available in Android?**
    - **LinearLayout**, **RelativeLayout**, **ConstraintLayout**, **FrameLayout**, **TableLayout**, **GridLayout**. 📐
72. **How does Toast work in Android?**
    - Displays short messages at the bottom of the screen. 🍞
73. **What are SharedPreferences in Android?**
    - Key-value storage for small data. 🔑
74. **ArrayAdapter in Android**
    - Binds data to views like ListView. 📊
75. **Features of SQLite Database in Android**
    - Lightweight, serverless, embedded database. 💽
76. **SQLite Database Operations in Android**

- Efficiently stores and retrieves structured data. 📥