Mobile Computing

Here are 100 questions with simple answers based on the topics you've listed:

Module 1: Introduction to Android and its Components

1. **Q: What is an Android application?**
   A: An Android application is a software application built to run on Android devices, developed using Android SDK.
2. **Q: What are the components of an Android application?**
   A: The main components are Activities, Services, Broadcast Receivers, and Content Providers.
3. **Q: What is the purpose of the AndroidManifest.xml file?**
   A: The AndroidManifest.xml file is used to declare the components of an Android app and to define essential information such as permissions.
4. **Q: What is an Activity in Android?**
   A: An Activity represents a single screen with a user interface in an Android app.
5. **Q: What is an Intent in Android?**
   A: An Intent is an abstract description of an operation to be performed, such as starting an Activity.
6. **Q: What are Fragments in Android?**
   A: Fragments are reusable portions of an Activity's user interface that can be combined in different ways to create a flexible UI.
7. **Q: What are Services in Android?**
   A: Services are components that run in the background to perform long-running operations without user interaction.
8. **Q: What are the different types of layouts in Android?**
   A: Types of layouts include LinearLayout, RelativeLayout, FrameLayout, ConstraintLayout, and TableLayout.
9. **Q: What is the purpose of Views in Android?**
   A: Views are the basic building blocks of UI components, such as buttons, text fields, and images.
10. **Q: How do you display a picture in Android?**
    A: A picture can be displayed using the ImageView widget in Android.
11. **Q: What is a menu in Android?**
    A: A menu is a user interface component that displays a list of options or actions, typically in the app bar or via a context menu.
12. **Q: How do you work with intents in Android?**
    A: You can create an Intent object, set its action and target, and then use startActivity() or startService() methods to launch it.
13. **Q: What are explicit and implicit intents?**
    A: Explicit intents specify the exact component to launch, while implicit intents declare an action and let the system find the appropriate component.

14. **Q: How do you launch a new Activity?**
    A: You launch a new Activity using the startActivity() method with an Intent object.

15. **Q: What is the difference between LinearLayout and RelativeLayout?**
    A: LinearLayout arranges its child views in a single direction, either vertically or horizontally, while RelativeLayout allows views to be positioned relative to each other.

Module 2: Basic Controls and UI Components

16. **Q: What is a TextView in Android?**
    A: A TextView is a UI component used to display text to the user.

17. **Q: What is a RadioButton in Android?**
    A: A RadioButton is a UI component that allows the user to select one option from a group of choices.

18. **Q: What is a CheckBox in Android?**
    A: A CheckBox is a UI component that allows the user to select or deselect an option.

19. **Q: What is an ImageButton in Android?**
    A: An ImageButton is a button that displays an image instead of text.

20. **Q: What is an EditText in Android?**
    A: EditText is a UI component used to capture user input in the form of text.

21. **Q: What is a Slider in Android?**
    A: A Slider is a UI component that allows the user to select a value from a range by sliding a handle.

22. **Q: How can you create a button in Android?**
    A: A button can be created using the <Button> tag in the XML layout file.

23. **Q: How do you set a listener for a button in Android?**
    A: You set a listener using button.setOnClickListener() method and defining an OnClickListener for the button.

24. **Q: How do you change the text of a TextView programmatically?**
    A: You can use textView.setText("New Text") to change the text.

25. **Q: What is the purpose of a Spinner in Android?**
    A: A Spinner is used to display a drop-down list of items for the user to select from.

26. **Q: How do you display a Toast message in Android?**
    A: You display a Toast message using Toast.makeText(context, "Message", Toast.LENGTH_SHORT).show().

27. **Q: How can you make a TextView editable?**
    A: You can make a TextView editable by changing it to an EditText widget.

28. **Q: What is the use of a ToggleButton?**
    A: A ToggleButton is a button that can be toggled between two states: ON or OFF.

29. **Q: How do you add a checkbox to a layout?**
    A: You add a checkbox by using <CheckBox> in the layout XML file.

30. **Q: How do you set default values for UI components?**
    A: Default values can be set in the XML file using attributes like android:text, android:checked, etc.

Module 3: Database Connectivity

31. **Q: What is SQLite?**
    A: SQLite is a lightweight database used for storing data in Android applications.
32. **Q: How do you perform a CRUD operation in SQLite?**
    A: You perform CRUD operations using methods like insert(), update(), delete(), and query().
33. **Q: What is SharedPreferences in Android?**
    A: SharedPreferences is a way to store simple key-value pairs in persistent storage.
34. **Q: How can you store data in internal storage?**
    A: Data can be stored in internal storage using FileOutputStream and FileInputStream.
35. **Q: How do you access external storage in Android?**
    A: External storage is accessed using getExternalStorageDirectory().
36. **Q: What is the purpose of Content Providers?**
    A: Content Providers manage access to a structured set of data, and they can be used for sharing data between applications.
37. **Q: How do you insert data into an SQLite database?**
    A: You insert data using the insert() method of SQLiteDatabase.
38. **Q: What is the difference between internal and external storage?**
    A: Internal storage is private to the app, while external storage is shared among apps and can be accessed by other applications.
39. **Q: How do you retrieve data from SQLite in Android?**
    A: Data is retrieved using the query() method of SQLiteDatabase.
40. **Q: How do you delete data from SQLite in Android?**
    A: You delete data using the delete() method of SQLiteDatabase.

Module 4: Graphics and Animation, Multimedia

41. **Q: How do you draw graphics in Android?**
    A: You draw graphics using the Canvas class and the draw methods.
42. **Q: What is the purpose of Animation in Android?**
    A: Animation in Android allows you to create visual effects by changing properties of UI components over time.
43. **Q: What are the types of animation in Android?**
    A: Types include property animation, view animation, and drawable animation.
44. **Q: How do you play audio in Android?**
    A: Audio can be played using the MediaPlayer class.
45. **Q: How do you play video in Android?**
    A: Video can be played using the VideoView class.

46. **Q: What is the purpose of the SurfaceView in Android?**
    A: SurfaceView is a specialized view for displaying video or other graphics content that can be rendered off the main UI thread.

47. **Q: How do you capture photos in Android?**
    A: Photos can be captured using the Camera API or Camera2 API.

48. **Q: What is SMS in Android?**
    A: SMS (Short Message Service) is a way to send text messages from one mobile device to another.

49. **Q: How do you send an email in Android?**
    A: Emails can be sent using Intent with the action Intent.ACTION_SENDTO and the mailto URI.

50. **Q: What is the use of the MediaPlayer class?**
    A: The MediaPlayer class is used to play audio and video files.

Module 5: Location-Based Services

51. **Q: How do you display a map in Android?**
    A: A map can be displayed using the GoogleMap class in the Google Maps API.

52. **Q: How do you get the location of the device in Android?**
    A: The device's location can be obtained using the LocationManager class or FusedLocationProviderClient.

53. **Q: What is geocoding in Android?**
    A: Geocoding is the process of converting an address into geographical coordinates (latitude and

longitude).

54. **Q: What is reverse geocoding in Android?**
    A: Reverse geocoding is the process of converting geographical coordinates (latitude and longitude) into an address.

55. **Q: How do you monitor the location of the device?**
    A: Location updates can be monitored using LocationListener or FusedLocationProviderClient.

56. **Q: What is the purpose of a location tracker in Android?**
    A: A location tracker monitors the device's movement and can provide updates about its location.

57. **Q: How do you calculate the distance between two locations in Android?**
    A: The distance can be calculated using the Location.distanceBetween() method.

58. **Q: How do you show the user's location on the map?**
    A: You can use GoogleMap.setMyLocationEnabled(true) to show the user's location on the map.

59. **Q: How do you move the map's camera to a specific location?**
    A: The camera can be moved using GoogleMap.moveCamera() with a LatLng object.

60. **Q: What is the FusedLocationProviderClient in Android?**
    A: It is a class used for obtaining location updates using the most accurate location providers.

Module 6: REST API Integration

61. **Q: What is an API in Android?**
    A: An API (Application Programming Interface) allows communication between different software applications.

62. **Q: How do you consume web services using HTTP in Android?**
    A: You use HttpURLConnection or OkHttp to make HTTP requests.

63. **Q: What is JSON?**
    A: JSON (JavaScript Object Notation) is a lightweight data interchange format that is easy for humans to read and write.

64. **Q: How do you parse JSON data in Android?**
    A: JSON data can be parsed using JSONObject and JSONArray classes.

65. **Q: What is AsyncTask in Android?**
    A: AsyncTask is used to perform background operations without blocking the main thread.

66. **Q: What is Retrofit in Android?**
    A: Retrofit is a type-safe HTTP client for Android that simplifies consuming REST APIs.

67. **Q: How do you make a network call using Retrofit?**
    A: You create a Retrofit instance, define an interface for the API endpoints, and make the network call asynchronously.

68. **Q: What is Volley in Android?**
    A: Volley is an HTTP library used for making network requests in Android.

69. **Q: What is OkHttp in Android?**
    A: OkHttp is an efficient HTTP client for Android used for making network requests.

70. **Q: How do you handle JSON in Android?**
    A: JSON can be handled using JSONObject and JSONArray classes to parse and extract data.

Module 7: Introduction to Dart and Flutter

71. **Q: What is Dart?**
    A: Dart is a programming language developed by Google, often used for building mobile, web, and desktop apps.

72. **Q: What is Flutter?**
    A: Flutter is an open-source UI framework for building natively compiled applications for mobile, web, and desktop using a single codebase.

73. **Q: What is the structure of a Dart program?**
    A: A Dart program consists of classes, functions, and methods, typically with a main() function as the entry point.

74. **Q: What are widgets in Flutter?**
    A: Widgets are the basic building blocks of a Flutter UI, representing elements like buttons, text, and images.

75. **Q: What are stateful and stateless widgets in Flutter?**
    A: Stateless widgets have no mutable state, while stateful widgets can change their state during their lifecycle.

76. **Q: What is the purpose of the build() method in Flutter?**
    A: The build() method returns a widget that represents the current state of the UI.

77. **Q: How do you navigate between screens in Flutter?**
    A: Navigation is done using Navigator.push() and Navigator.pop() methods.

78. **Q: How do you create a layout in Flutter?**
    A: Layouts are created using various widgets such as Column, Row, Container, and Stack.

79. **Q: What is the difference between hot reload and hot restart in Flutter?**
    A: Hot reload preserves the app state, while hot restart resets the app state and reloads the entire app.

80. **Q: How do you style widgets in Flutter?**
    A: Widgets can be styled using the style property, such as TextStyle for text and BoxDecoration for containers.

Module 8: Data Handling

81. **Q: What is JSON in Flutter?**
    A: JSON is a format used to store and transfer data, often used in Flutter to communicate with web services.

82. **Q: How do you parse JSON in Flutter?**
    A: JSON is parsed using the dart:convert package's jsonDecode() function.

83. **Q: What is SQLite used for in Flutter?**
    A: SQLite is used for storing structured data locally in Flutter applications.

84. **Q: How do you handle data in Flutter forms?**
    A: Forms are handled using TextEditingController and Form widget with validation.

85. **Q: How do you style widgets in Flutter?**
    A: Widgets are styled using properties like padding, margin, decoration, and color.

86. **Q: What is the purpose of the setState() method in Flutter?**
    A: setState() is used to notify the framework that the widget's state has changed and needs to be rebuilt.

87. **Q: What is the TextFormField widget in Flutter?**
    A: TextFormField is a widget that provides a form-like input field with built-in validation.

88. **Q: How do you manage state in Flutter?**
    A: State can be managed using setState(), Provider, or other state management techniques.

89. **Q: How do you handle database operations in Flutter?**
    A: Database operations are handled using packages like sqflite for SQLite or moor for local storage.

90. **Q: What is the difference between ListView and GridView in Flutter?**
    A: ListView displays a scrollable list of items, while GridView displays a scrollable grid of items.

Module 9: Case Study on iOS App Development

91. **Q: What is Swift programming?**
    A: Swift is a programming language developed by Apple for building iOS, macOS, and watchOS applications.

92. **Q: What is Objective-C?**
    A: Objective-C is an object-oriented programming language used for iOS and macOS development before Swift.

93. **Q: What is the role of the AppDelegate in iOS?**
    A: The AppDelegate manages app lifecycle events such as app launch, background, and foreground states.

94. **Q: How do you create a simple view controller in Swift?**
    A: You create a view controller by subclassing UIViewController and overriding lifecycle methods like viewDidLoad().

95. **Q: What is Auto Layout in iOS?**
    A: Auto Layout is a system used to define the position and size of UI elements relative to each other and their container.

96. **Q: How do you manage navigation in iOS?**
    A: Navigation is managed using UINavigationController to push and pop view controllers.

97. **Q: What is the purpose of UITableView in iOS?**
    A: UITableView is used to display a scrollable list of items, similar to ListView in Android.

98. **Q: How do you handle JSON in iOS?**
    A: JSON is handled using JSONSerialization for parsing and URLSession for networking.

99. **Q: What is the difference between UIView and UIViewController in iOS?**
    A: UIView represents a visual element, while UIViewController manages a view and its interactions.

100. **Q: How do you handle app permissions in iOS?**
    A: App permissions are handled using the Info.plist file, where you define required permissions such as camera or location access.

Implementation

To help you complete each of these Android and Flutter programs, I'll break down the tasks in simple steps for you:

1. Android Program using Various UI Components

Objective: Create a layout with different UI components like EditText, TextView, Button, etc. **Steps:**

- Open Android Studio and create a new project.
- In activity_main.xml, add various UI components:
- <EditText
- android:id="@+id/editText"
- android:layout_width="wrap_content"
- android:layout_height="wrap_content"
- android:hint="Enter text" />
-
- <Button
- android:id="@+id/button"
- android:layout_width="wrap_content"
- android:layout_height="wrap_content"
- android:text="Click Me" />
-
- <TextView
- android:id="@+id/textView"
- android:layout_width="wrap_content"
- android:layout_height="wrap_content"
- android:text="Hello World" />
- In MainActivity.java, find the views and add click logic:
- Button button = findViewById(R.id.button);
- button.setOnClickListener(v -> {
- EditText editText = findViewById(R.id.editText);
- TextView textView = findViewById(R.id.textView);
- textView.setText(editText.getText().toString());
- });

2. Android Program using Different Layouts and Views

Objective: Use multiple layout types (LinearLayout, RelativeLayout, etc.). **Steps:**

- In activity_main.xml, use different layouts:
- <LinearLayout
- android:layout_width="match_parent"
- android:layout_height="match_parent"
- android:orientation="vertical">
-
- <TextView
- android:layout_width="wrap_content"
- android:layout_height="wrap_content"
- android:text="Linear Layout"/>
-
- </LinearLayout>

Then, switch to RelativeLayout or ConstraintLayout to experiment with different designs.

3. Android Program Based on Intents

Objective: Use intents to open another activity. **Steps:**

- Create two activities (MainActivity and SecondActivity).
- In MainActivity.java, use an intent to open the second activity:
- Intent intent = new Intent(MainActivity.this, SecondActivity.class);
- startActivity(intent);

4. Android Program for Notifications and Alert Box

Objective: Show a notification and an alert box. **Steps:**

- **For Notification:**
- NotificationCompat.Builder builder = new NotificationCompat.Builder(this, "ChannelID")
- .setContentTitle("My Notification")
- .setContentText("Hello, this is a notification")
- .setSmallIcon(R.drawable.ic_notification);
-
- NotificationManagerCompat notificationManager = NotificationManagerCompat.from(this);
- notificationManager.notify(1, builder.build());

- **For Alert Box:**
- new AlertDialog.Builder(this)
- .setMessage("Are you sure?")
- .setPositiveButton("Yes", null)
- .setNegativeButton("No", null)
- .show();

5. Android Program to Perform CRUD Operation Using SQLite DB

Objective: Use SQLite to create, read, update, and delete data. **Steps:**

- Create a SQLite database helper class that extends SQLiteOpenHelper.
- Define methods to add, update, delete, and fetch data from the database.

```
public class DBHelper extends SQLiteOpenHelper {
   @Override
   public void onCreate(SQLiteDatabase db) {
      db.execSQL("CREATE TABLE my_table(id INTEGER PRIMARY KEY, name TEXT)");
   }

   public void addData(String name) {
      SQLiteDatabase db = this.getWritableDatabase();
      ContentValues values = new ContentValues();
      values.put("name", name);
      db.insert("my_table", null, values);
   }
}
```

6. Android Program Using Shared Preferences, Internal and External Storage

Objective: Store data in SharedPreferences and internal/external storage. **Steps:**

- **SharedPreferences:**
- SharedPreferences sharedPreferences = getSharedPreferences("MyPrefs", MODE_PRIVATE);
- SharedPreferences.Editor editor = sharedPreferences.edit();
- editor.putString("key", "value");
- editor.apply();
- **Internal Storage:**

- FileOutputStream fos = openFileOutput("data.txt", Context.MODE_PRIVATE);
- fos.write("Hello".getBytes());
- fos.close();
- **External Storage:**
- File file = new File(Environment.getExternalStorageDirectory(), "data.txt");
- FileOutputStream fos = new FileOutputStream(file);
- fos.write("Hello".getBytes());
- fos.close();

7. Android Program to Work with Graphics and Animation

Objective: Use Canvas for drawing or create simple animations. **Steps:**

- **For Graphics:**
- @Override
- protected void onDraw(Canvas canvas) {
- super.onDraw(canvas);
- Paint paint = new Paint();
- paint.setColor(Color.RED);
- canvas.drawCircle(50, 50, 30, paint);
- }
- **For Animation:**
- ObjectAnimator animator = ObjectAnimator.ofFloat(view, "translationX", 0f, 500f);
- animator.setDuration(1000);
- animator.start();

8. Android Program to Work with Google Maps and Locations

Objective: Display Google Maps and get current location. **Steps:**

- Add the Google Maps API key to AndroidManifest.xml.
- Use GoogleMap in your activity to show a map:
- GoogleMap map = ((MapFragment) getFragmentManager().findFragmentById(R.id.map)).getMap();
- Use FusedLocationProviderClient to get the current location:
- FusedLocationProviderClient fusedLocationClient = LocationServices.getFusedLocationProviderClient(this);
- fusedLocationClient.getLastLocation()

- .addOnSuccessListener(this, location -> {
- // Use location
- });

9. Android Program to Work with Images and Videos

Objective: Display images and videos in the app. **Steps:**

- **For Images:** Use ImageView to display an image:
- <ImageView
- android:id="@+id/imageView"
- android:layout_width="match_parent"
- android:layout_height="wrap_content"
- android:src="@drawable/sample_image"/>
- **For Videos:** Use VideoView to play videos:
- <VideoView
- android:id="@+id/videoView"
- android:layout_width="match_parent"
- android:layout_height="wrap_content"/>

And in Activity:

VideoView videoView = findViewById(R.id.videoView);

videoView.setVideoURI(Uri.parse("android.resource://" + getPackageName() + "/" + R.raw.sample_video));

videoView.start();

10. Android Program Based on RestAPI

Objective: Make network calls to a REST API. **Steps:**

- Use Retrofit or HttpURLConnection to make API requests.
- Example using HttpURLConnection:
- URL url = new URL("https://api.example.com/data");
- HttpURLConnection urlConnection = (HttpURLConnection) url.openConnection();
- InputStream in = new BufferedInputStream(urlConnection.getInputStream());

11. Flutter Program Using Layout Widgets and State Management

Objective: Use Flutter widgets and manage state. **Steps:**

- Create a layout using widgets like Container, Column, and Row.

- Column(
- children: <Widget>[
- Text('Hello Flutter'),
- ElevatedButton(
- onPressed: () {},
- child: Text('Click Me'),
-),
-],
-)
- For state management, use setState():
- setState(() {
- _counter++;
- });

12. Flutter Program to Work with SQLite

Objective: Use SQLite for local storage. **Steps:**

- Add dependencies for SQLite in pubspec.yaml.
- Create a database helper to handle CRUD operations.

final db = await openDatabase('my_db.db');

await db.insert('my_table', {'name': 'John'});

13. Flutter program based on RestAPI

- Set up a Flutter project using flutter create.
- Add dependencies (like http) in pubspec.yaml.
- Create a model class to represent the data you will fetch from the API.
- Make an HTTP request using the http package to fetch data from the API.
- Parse the JSON response and convert it into model objects.
- Display the fetched data in the app using a FutureBuilder.