**1. Programming Language Fundamentals (Java/Kotlin)**

* basic language constructs: variables, operators, loops, conditions;
* collections (List, Set, Map);
* OOP: encapsulation, inheritance, polymorphism, abstraction;
* exceptions and handling (try-catch, finally);
* working with threads (Threads, Coroutines);
* lambda expressions and functional programming features (Kotlin).

**2. Android SDK and Android Architecture**

* lifecycle of Activity and Fragment;
* Intent, BroadcastReceiver, Service, ContentProvider;
* View, ViewGroup, RecyclerView, ConstraintLayout;
* working with resources (res, strings.xml, styles.xml);
* data storage: SharedPreferences, Room, DataStore;
* API interaction: Retrofit, OkHttp;
* working with asynchronous tasks: AsyncTask, Handler, LiveData, Coroutines.

**3. Multithreading Basics**

* Thread, Runnable, ExecutorService;
* Handler, Looper, MessageQueue;
* RxJava or Kotlin Coroutines.

**4. Android Application Architecture**

* MVVM, MVP, Clean Architecture;
* ViewModel, LiveData;
* Dependency Injection (Dagger, Hilt, Koin).

**5. Working with APIs and Databases**

* REST API (Retrofit, OkHttp);
* JSON parsing (Gson / Moshi);
* local databases: SQLite, Room.

**6. Jetpack and Modern Development Tools**

* ViewModel, LiveData;
* Navigation Component;
* WorkManager;
* DataBinding / ViewBinding;
* Paging, RecyclerView.

**7. Git and Version Control Systems**

* basic Git commands (clone, pull, push, commit, merge, rebase);
* working with branches (branch, checkout, merge).

**8. CI/CD and Basic DevOps Knowledge**

* working with Gradle;
* setting up ProGuard;
* automated builds (GitHub Actions, Jenkins).

**9. Testing (Optional but a Plus)**

* unit testing (JUnit, Mockito);
* UI testing (Espresso);
* integration testing.

**10. Publishing Applications**

* preparing the app for release;
* uploading to Google Play (preparing .apk or .aab);
* working with Firebase Analytics and Crashlytics.

**What’s Important for a Junior?**

* knowledge of Android SDK basics;
* strong grasp of Kotlin or Java;
* understanding of key development patterns;
* working with REST APIs and databases.

**What’s Important for a Middle?**

* strong understanding of MVVM and Clean Architecture;
* knowledge of multithreading (Coroutines, RxJava);
* experience with Jetpack libraries;
* writing tests (unit, integration, UI);
* experience with CI/CD and Gradle.