

# Kotlin Quiz

Kindly attempt this questions before our session on 22nd of June. We shall review it together.

\* Indicates required question

---

1. Email \*

---

2. What file is the entry point for an Android application? \*

1 point

*Mark only one oval.*

- ☐ MainActivity.kt
- ☐ AndroidManifest.xml
- ☐ build.gradle.kts
- ☐ Main.kt

3. What is the use of the res directory in an Android project? \*

1 point

*Mark only one oval.*

- ☐ To store the source code
- ☐ To store resource files like layouts, strings, and images
- ☐ To store third-party libraries
- ☐ To store configuration files

4. What is the purpose of Gradle in an Android project? \*

1 point

*Mark only one oval.*

- ☐ To manage UI components
- ☐ To handle HTTP requests
- ☐ To automate the build process
- ☐ To manage user authentication

5. What is the purpose of the build.gradle file in an Android project? \*

1 point

*Mark only one oval.*

- ☐ It defines the project-wide settings and configuration
- ☐ It contains the Kotlin source code for the project
- ☐ It manages user interface components
- ☐ It handles database transactions

6. How does Android Studio manage dependencies in an Android project, and where are these configurations typically found?

\* 1 point

*Mark only one oval.*

- ☐ Dependencies are managed through the AndroidManifest.xml file
- ☐ Dependencies are declared in the build.gradle files
- ☐ Dependencies are handled by the Android SDK manager
- ☐ Dependencies are controlled through the proguard-rules.pro file

7. How does Android Studio support Unit Testing and UI Testing for Android applications? \* 1 point

*Mark only one oval.*

- ☐ Android Studio includes built-in frameworks and tools like Espresso for UI testing and JUnit for unit testing
- ☐ Android Studio uses Firebase for unit testing
- ☐ Android Studio integrates with third-party testing tools only
- ☐ Android Studio manages database transactions for testing purposes

8. What is the advantage of using an Android emulator for app development? \* 1 point

*Mark only one oval.*

- ☐ Emulators provide faster performance compared to physical devices
- ☐ Emulators allow developers to test apps on a wide range of device configurations
- ☐ Emulators have longer battery life
- ☐ Emulators are more reliable for network operations

9. How does Android Studio facilitate testing on physical devices during app development? \* 1 point

*Mark only one oval.*

- ☐ It automatically connects to any physical device within the vicinity
- ☐ It provides drivers for various device manufacturers
- ☐ It uses Bluetooth for connecting to physical devices
- ☐ It enables USB debugging and deployment directly to connected devices

10. What are some limitations of using an Android emulator compared to a physical device for testing? \* 1 point

*Mark only one oval.*

- ☐ Emulators do not support network operations
- ☐ Emulators cannot simulate real-world performance accurately
- ☐ Emulators do not support UI components
- ☐ Emulators have limited storage capacity

11. Explain the process of setting up and running an Android application on a physical device from Android Studio. \* 1 point

*Mark only one oval.*

- ☐ Connect the device via USB, enable USB debugging in developer settings, select the device from the Run/Debug configurations, and click Run
- ☐ Install drivers from the Android SDK manager, connect via Wi-Fi, and deploy the app using the Quick Deploy feature
- ☐ Pair the device with Android Studio using Bluetooth, configure app settings, and use the Remote Control feature to deploy
- ☐ None of the above

12. What is the purpose of debugging in Android development? \* 1 point

*Mark only one oval.*

- ☐ To optimize app performance
- ☐ To test app functionality on physical devices
- ☐ To identify and fix errors or bugs in the code
- ☐ To manage app permissions

13. Explain the role of breakpoints in debugging with Android Studio. \* 1 point

*Mark only one oval.*

- ☐ Breakpoints are used to terminate the app execution
- ☐ Breakpoints are markers set in the code to pause execution and inspect variables and state
- ☐ Breakpoints handle network requests
- ☐ Breakpoints manage user authentication

14. What is Logcat in Android Studio, and how is it used during debugging? \* 1 point

*Mark only one oval.*

- ☐ Logcat displays real-time system log messages from the device or emulator, helping developers monitor app behavior and diagnose issues
- ☐ Logcat manages app resources
- ☐ Logcat optimizes UI responsiveness
- ☐ Logcat handles database transactions

15. Explain what Coroutines are and their use in Android. \* 1 point

*Mark only one oval.*

- ☐ They are methods for drawing UI components
- ☐ They are a concurrency design pattern used to simplify code that executes asynchronously
- ☐ They are tools for managing app resources
- ☐ They are techniques for handling multiple users

16. What is the difference between val and var? \*

1 point

*Mark only one oval.*

- ☐ Variables declared with val can only access const members
- ☐ Variables declared with val are final, those with var are not
- ☐ var is scoped to the nearest function block and val is scoped to the nearest enclosing block
- ☐ Variables declared with var are final, those with val are not

17. What does a data class not offer? \*

1 point

*Mark only one oval.*

- ☐ An auto-generated toString() method
- ☐ Automatic conversion from/to JSON
- ☐ Auto-generated hashCode() and equals() methods
- ☐ A generated copy(...) method, to create copies of instances

18. What does this code do? foo()() \*

1 point

*Mark only one oval.*

- ☐ Creates a two-dimensional array
- ☐ Fails compilation
- ☐ Calls a function which is returned by the call of foo
- ☐ Calls foo asynchronously

19. Which of the following is a feature of Kotlin that allows you to write expressive and concise code by removing boilerplate code? \* 1 point

*Mark only one oval.*

- ☐ Data classes
- ☐ Abstract classes
- ☐ Static classes
- ☐ Utility classes

20. Write a Kotlin function that demonstrates null safety using the safe call operator and the Elvis operator. \* 1 point

---

21. Write a Kotlin function that uses the when expression to return a grade based on a score input \* 1 point

```
fun getGrade(score: Int): String {  
  
}
```

---

22. Create a higher-order function in Kotlin that takes a function as a parameter and applies it to the number 10. \* 1 point

---

23. Kotlin supports both functional and object-oriented programming paradigms. \* 1 point

*Mark only one oval.*

- ☐ True
- ☐ False

24. In Kotlin, all classes are final by default unless explicitly marked with the `open` keyword. \* 1 point

*Mark only one oval.*

☐ True

☐ False

25. Which annotation is used to denote a test method in JUnit 5? \* 1 point

*Mark only one oval.*

☐ @TestMethod

☐ @Test

☐ @TestCase

☐ @JUnitTest

26. Write a simple JUnit 5 test function in Kotlin that checks if the sum of two numbers is correct. \* 0 points

---

27. Write a Kotlin test using `assertFailsWith` to check that a function throws an `IllegalArgumentException` when given an invalid argument. \* 1 point

---

---

This content is neither created nor endorsed by Google.

Google Forms



