

# Kotlin - Enhanced Android Development

Hemalatha<sup>1</sup>, Arun Kumar<sup>2</sup>, Dinesh<sup>3</sup>, Mukesh<sup>4</sup>

<sup>1</sup>Assistant Professor, Dept of Computer Science and Engineering

<sup>2,3,4</sup>Dept of Computer Science and Engineering

<sup>1,2,3,4</sup>Sri Shakthi Institute of Engineering and Technology, Coimbatore

**Abstract-** There are many languages to create an Android application. However, Kotlin is also one of the programming languages to create an Android application in which it supports Android Studio for the development of android application. Our aim is to make sure which language is more suitable for development of android application and to show that it is the best procedural programming language. This language saves a lot of time and it is more efficient and easy to code.

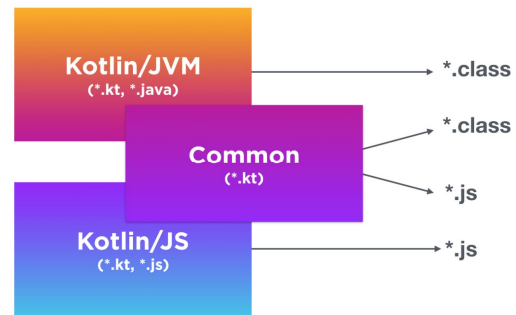
**Keywords-** Kotlin, Android application, Java.

## I. INTRODUCTION

Kotlin is one of the emerging and efficient programming language for many multi-platform applications. It helps in building many applications like Android, Server Side, Web. The Kotlin/Native is the technology that compiles Kotlin to native binaries that runs without any Virtual Machines. Kotlin compiles to JVM byte code or JavaScript and it runs on Java Virtual Machine (JVM). It enables the programmers to easily migrate the existing Java Code to Kotlin even for complex programs that runs millions of lines of code.

As Kotlin is developed based on Java, it can support the usage of existing Java libraries and frameworks. It imposes no runtime overhead as the standard library is small and tight. Kotlin allows to keep using the productivity environment tools and if the IntelliJ IDE is used, then interoperability is entirely seamless where Kotlin codes can be refactored, navigated and auto completed as if the Kotlin code was Java and vice versa.

Kotlin is starting to become popular with Android Developers as it has many new features that Java does not have. Some of them are Null Safety, Lambda Expressions, functional programming and so on. Kotlin targets Java 6 and so it can be used even if the deployment makes upgrading to a newer JVM difficult.



## II. FEATURES

- 1. Functional Programming:** Kotlin supports functional programming with zero-overhead lambdas and ability to do mapping, folding over standard Java collections. It also includes higher-order functions. A higher-order function is a function that takes functions as parameters, or returns a function. For a function to accept another function as a parameter, we have to specify a function type for that parameter.
- 2. Null Safety:** In Java, the most common cause for application crash is that accessing a member of a null reference will result in NPE called as Null Pointer Exception. In Kotlin, it is packed with null safety which lets the compiler systematically flag potential null pointer dereferences. Kotlin's type system is aimed to eliminate the Null Pointer Exception from the code. In Kotlin, the type system easily distinguishes between the nullable and non-null references.
- 3. Lambda Expressions:** The Lambda Expressions is called as "Function Literal". It is that a function that is not declared, but passed immediately as an expression. A lambda expression is always surrounded by curly braces, parameter declarations in the full syntactic form go inside curly braces and have optional type annotations, the body goes after an -> sign.

```
val sum = { x: Int, y: Int -> x + y }
```

- 4. Operator Overloading:** In Java, Operator Overloading is not supported where + is used to concatenate two strings. Operator overloading are very much helpful in making the

code very less and more readable. In Kotlin, there is a support for operator overloading. Several set of operators are employed with variety of functionalities to be performed. The basic functionality of operator's expressions are translated to respective functions.

Expression	Translated to
<code>a + b</code>	<code>a.plus(b)</code>

```
data class Doggo(val breed: String, val barksPerHour: Int) {
    ... constructor() : this("Unidentified", -1)
}

data class Catto(val breed: String = "Unidentified", val meowsPerHour: Int = -1)
```

### III. PROS AND CONS OF KOTLIN

**PROS OF KOTLIN:** One of the strong points of Kotlin is its **100% Java interoperability**. The language works well with Java itself, including all related tools and frameworks, which provides a rich ecosystem. Moreover, this makes it possible to gradually migrate to Kotlin, or to successfully use both languages within the same project.

With a more compact and clear codebase, Kotlin allows for **less errors and more stable code in production**. The compiler detects all possible errors at compile time, not at run time. This makes Kotlin a safer alternative for Java.

The language is compact, clear and efficient, it has a concise and intuitive syntax. Kotlin significantly increases the team productivity. It takes very less time to write and deploy new code and also contributes to better code maintainability at scale.

**CONS OF KOTLIN:** Some developers report slower compilation speed as one of the main drawbacks of Kotlin. However, the results of multiple tests are mixed: In some cases, Kotlin beats Java in compilation speed, but sometimes it is significantly slower than its predecessor.

As Kotlin is still relatively new to most of the developers, it might be hard to find experienced professionals in this domain.

The Kotlin Standard Library and runtime will increase the size of your .apk. While this only equates to

around 800KB, if your application is already on the large side then that extra 800KB may tip it over the edge and make users think twice before downloading your app.



### VII. CONCLUSION

Considering Kotlin with Java it better to prefer Kotlin than Java. Because Kotlin is easy to code and it is a best procedural programming language. Kotlin saves more time and it is more efficient.

### REFERENCES

- [1] <https://kotlinlang.org/>
- [2] <https://developer.android.com/kotlin/index.html>
- [3] <https://kotlinlang.org/docs/tutorials/>
- [4] [https://en.wikipedia.org/wiki/Kotlin\\_\(programming\\_language\)](https://en.wikipedia.org/wiki/Kotlin_(programming_language))