In many ways, Kotlin designed by JetBrains has become the preferred language for the Android development industry. While it cannot outdo Java in terms of its popularity, many developers emphasise Kotlin will soon become the dominant platform for Android development.

When it comes to its usability, Kotlin has a bunch of tools and frameworks that work with Java. Also, **its advanced compiler can check the errors both during compile time and run time**. It also brings down the number of lines of code significantly. Kotlin is officially supported by Google for mobile development on Android since the release of Android Studio 3.0 in October 2017.

Here are 5 reasons why developers like Kotlin over Java and think that it will overtake the later:

**1. Shorter program for the same task**

Kotlin is a statically-typed language which is very easy to read and write. It has a much simpler and shorter code than Java’s code for the same problem. As this makes the language more human-readable, it becomes easy to debug. **Kotlin’s code is much smaller and streamlines the programming process, in comparison to Java**. This is partly because of Kotlin’s slick IDE.

**2. Easy Code**

Kotlin programs do not need semicolons in their program. This makes the programs easy to read and understand. They also have smart casts and string templates. **Java is not a succinct language. Such a language code increases the chances of bugs**. The code being in a concise language means fewer chances of both runtime and compile time errors. Kotlin gives a simple way to use mutable and immutable declarations for different data structures.

**3. Java Compatibility**

Kotlin can easily exchange and use information from Java in a number of ways. This is one of the most powerful advantages of Kotlin**. Java and Kotlin code can co-exist in the same project**. Kotlin plays well with the Java programming language. Moreover, a number of Java libraries can be used in Kotlin projects, making it even more compatible.

Not just the libraries but plenty of frameworks from Java are compatible with Kotlin, including some advanced frameworks**. You can use Kotlin from Java without any major steps like converting an entire project to Kotlin.** This is a huge advantage to developers since they do not really have to drastically learn a new language. Anyone who knows Java will be familiar with and be able to code in Kotlin. Translating Java to Kotlin is easy. In IntelliJ or Android Studio is the only thing required as a Kotlin plugin to translate Java to Kotlin.

**4. Eliminating Null References:**

**One of the biggest advantages of Kotlin over Java is the null references**. This null reference, referred by Sir Tony Hoare, a British computer scientist, as The Billion Dollar Mistake. Accessing a member of a null reference result in a null reference exception. This is one of the major drawbacks of Java in which it is called a NullPointerException or NPE. Kotlin’s type system is aimed to eliminate NullPointerException from the code. The only possible causes of NPE’s may be:

**1. An explicit call to throw NullPointerException();**

**2. Usage of the !! operator that is described below;**

**3. Some data inconsistency with regard to initialization, such as when:**

- An uninitialized this available in a constructor is passed and used somewhere (“leaking this”);

- A superclass constructor calls an open member whose implementation in the derived class uses uninitialized state;

**4. Java interoperation:**

- Attempts to access a member on a null reference of a platform type;

- Generic types used for Java interoperation with incorrect nullability, e.g. a piece of Java code might add null into a Kotlin MutableList<String>, meaning that MutableList<String?> should be used for working with it;

- Other issues caused by external Java code.

**5. Solution To Some Of Java’s Flaws**

Plain old Java has a number of flaws. It inhibits the very famous problem of null pointer. Kotlin attempted to solve these hurdles created by Java. **It has adopted things from a number of languages like C# and mainly from Scala, to overcome difficulties of Java**. It contains instances from the language of Pascal and is considered as very influential in the development of Kotlin. Elements like parameter lists and variable declarations with the data type following a variable could also be in Kotlin.

Kotlin removes the boilerplate code, greatly eliminates possible errors. It possesses features like delegations, late initializations. **It also addresses type safety in lists, which is a major problem in Java**. It was very easy to type a wrongly typed variable to a list before generics came along. This would typically lead to blowing up during the run time because the compiler does not detect it.