CSC 533 Kotlin

Harrison Nordmeyer
Becca Borgmeier
Sam Herdlick

Kotlin

History

- Developed by JetBrains in 2016
  - Created by Andrey Breslav
  - Started in 2010 (first commit)

- Current Version 2.3.20
- Open-sourced and backwards compatable as of version 1.0
- In 2019, it became Androids official programming language

Design goals

- increase productivity
  - Better readability
  - Safety for the programmer
  - Interoperable with Java
- does not compromise on Java’s compilation speed

- a "better language" than Java, while still fully interoperable with Java code, allowing companies to make a gradual migration from Java to Kotlin.
  - Easy to switch from Java
  - Easier to read then Java
  - Interoperable with Java

Language features (1)

- Typing & Safety
  - Static typing
  - Null safety: String vs. String?
- Data Types
  - Primitive types (numbers, characters, booleans), other: strings, arrays

```kotlin

    // ── NULL SAFETY ──────────────────────────────────────────
    var name: String? = null       // "?" means this variable CAN be null

    // Safe call operator — only runs if name is not null
    println(name?.length)          // prints: null

    name = "Kotlin"
    println(name?.length)          // prints: 6

    // Elvis operator — provides a default if null
    val displayName = name ?: "Guest"
    println("Hello, $displayName!")
}

```

Language features (2)

- Control flow
  - Conditionals: if, when
    - Expression
  - Loops: for, while, do-while
  - Jumps: break, continue, return
  - Exception handling: try-catch-finally

```kotlin
fun main() {

    // ── IF / ELSE ──────────────────────────────────────────
    val temperature = 72
    val weather = if (temperature > 80) "Hot" else if (temperature > 60) "Nice" else "Cold"
    println("Weather: $weather")


    // ── WHEN (Kotlin's switch statement) ──────────────────
    val day = "Monday"
    when (day) {
        "Monday", "Tuesday", "Wednesday", "Thursday", "Friday" -> println("$day: Weekday")
        "Saturday", "Sunday" -> println("$day: Weekend")
        else -> println("Unknown day")
    }


    // ── FOR LOOP ──────────────────────────────────────────
    for (i in 1..5) {
        println("Count: $i")
    }


    // ── WHILE LOOP ──────────────────────────────────────────
    var count = 3
    while (count > 0) {
        println("Countdown: $count")
        count--
    }
```

Language features (3)

- Memory Management
  - Automatic garbage collection
- Abstraction support
  - Classes, interfaces
  - Functions (including higher-order functions and lambdas)
- Modern Style
  - Object oriented with functional aspects
- Concise Syntax

How features relate to Design Goals

- Readability (type inference & data classes for conciseness)
- Safety (null safety, static typing to prevent runtime errors)
- Interoperability & Pragmatism
  - Runs on JVM
  - Java interoperability

JAVA

```java
Public class Animal {
    Private int legs;

    public Animal(int legs) {
        this.legs = legs;
    }

    public int getLegs() {
        return legs;
    }

    public void setLegs(int legs) {
        this.legs = legs;
    }
}
```

KOTLIN

```kotlin
data class Animal(val legs: Int)  1 Usage
```

Kotlin Specific Features

- Null safety
- Flexible data classes
- Extension functions (easier to add functionality to existing classes)
- Coroutines (simpler way to handle concurrency)

Similar to Java but **safer, cleaner**, and **easier to write**

```kotlin
import kotlin.concurrent.thread

fun main() {
    repeat(50_000) {
        thread {
            Thread.sleep(5000L)
            print(".")
        }
    }
}
```

Kotlin Coroutines

```kotlin
suspend fun printPeriods() = coroutineScope { // this: CoroutineScope
    // Launches 50,000 coroutines that each wait five seconds, then print a period
    repeat(50_000) {
        this.launch {
            delay(5.seconds)
            print(".")
        }
    }
}
```

Real World Use

- Preferred for Android development
  - Google's official language since 2019, replacing Java and C++
- Widely adopted across the industry
  - Major companies like McDonald's, AWS, Adobe, and Google use it for its cross-platform development, expressiveness, and structured concurrency
- Null safety and smart cases
  - Reduces common errors, reliable choice for production code

Sources

- https://kotlinlang.org/spec/introduction.html
- https://www.educative.io/courses/kotlin-crash-course-for-programmers/kotlin-overview-principles-and-goals
- *YouTube*. https://www.youtube.com/watch?v=xT8oP0wy-A0. Accessed 23 Apr. 2026.
- https://medium.com/@rohinideshmane.21/introduction-to-kotlin-5f39b31610e0