Programming in Kotlin

Syllabus

### Prerequisites

Basic programming knowledge

### Duration

1 semester

### Recommendations for preparation

Students are expected to have a working programming environment. This may include any IDEs that support Kotlin, such as IntelliJ IDEA. Free educational licenses are available for download [here](https://www.jetbrains.com/community/education/#students).

### Description

In this course students will learn about the Kotlin programming language – a modern, powerful, and expressive language that is used for various purposes, from Android development all the way through to web development and data science. Students will learn how to apply Kotlin to solve practical software development problems and will learn about data types, variables and control flow, functions, object-oriented programming, exception handling, collections and generics, lambdas, and higher-order functions. They will also learn about various key features of Kotlin such as null safety, extension functions, and coroutines. At the end of the course students will study build systems, using Gradle as an example, as well as explore compilation techniques and how the Kotlin K2 compiler works.

### Contents

1. Introduction in Kotlin

2. Object-oriented programming

3. Generics

4. Collections

5. Functional programming

6. Build systems

7. Parallel and concurrent programming

8. Asynchronous programming

9. Java Virtual Machine and the Kotlin compiler (advanced topic)

10. Exceptions (optional topic)

11. Testing (optional topic)

**Assessment resources for educators:**

Quizzes

* For each topic

Homework assignments

* Homework 1, Alias game, topics 1-2
* Homework 2, Implementation of a Balanced Search Tree, topics 3-5
* Homework 3, fixing a broken Gradle build, topic 6
* Homework 4, creating a non-blocking UI simulator, topics 7-8

Tests

* Test 1, after topics 1-6
* Test 2, after topics 7-8
* Final test, after topic 9

### Goals

* Provide students with a solid foundation in the Kotlin programming language.
* Teach students how to apply Kotlin to solve practical problems in software development.
* Enable students to write efficient, readable, and maintainable Kotlin code.
* Familiarize students with key features of Kotlin, such as null safety, extension functions, and coroutines.
* Give students a deeper understanding of the fundamental concepts of computer science, such as concurrent computations and how they can be applied to software development in Kotlin.

### Intended learning outcomes

Upon completion of this module, students will be able to:

* Use the unique features of Kotlin to write readable, maintainable, and expressive code.
* Use Kotlin to solve practical problems in software development.
* Write efficient and optimized Kotlin code.
* Use the Gradle build system.
* Understand how the Kotlin compiler works.

### Recommended materials

— Roman Elizarov, Svetlana Isakova, Sebastian Aigner, and Dmitry Jemerov: [Kotlin in Action](https://www.manning.com/books/kotlin-in-action-second-edition), Second Edition, Manning Publications, 2022.

— [Kotlin Documentation](https://kotlinlang.org/docs/home.html)

— [Kotlin Onboarding: Introduction](https://plugins.jetbrains.com/plugin/21067-kotlin-onboarding-introduction)

— [Kotlin Onboarding: Object-Oriented programming](https://plugins.jetbrains.com/plugin/21913-kotlin-onboarding-object-oriented-programming)