

Introduction to Programming

7th Weekly Note (E17, Week 43)

Deadline

Remember the project deadline on October 27!

Reading for Week 43

- **Obligatory:** Chapter 1 and Sections 2.1–2.3, 2.5–2.6, 3.1, 3.4–3.5, and 4.2–4.4 of “Introduction to Programming Using Java”
- *Supplementary:* Chapters 1–2, 4–7, and 9 of “Think Java”

Lecture: Monday, October 23, 12-14 (U140)

After a look at a mobile app programmed in Java, we will learn how to express the basic constructs of imperative programming languages in Java. We start by learning about variables, static type declarations, assignments, serial execution and conditional execution.

Lecture: Thursday, October 26, 14-16 (U140)

We will continue to learn about function declarations, recursion, and iteration. Then we continue with type casts and exceptions.

Labs: see detailed schedule on course home page

A compiler is a program that translates a text representation of a program written in a programming language into a more machine-readable format. While this step is implicitly transformed by Python’s interpreter `python`, for most language this has to be done, before the program can be executed. Files that contain Java code always *have* to end with `.java`, as otherwise the compiler will refuse to work.

To compile the Java program described in the text file `HelloWorld.java` into the machine-readable program `HelloWorld.class`, use the command `javac HelloWorld.java`. To execute the compiled program, use the command `java HelloWorld` that executes the file `HelloWorld.class`. Note that the ending `.class` has to be omitted.

Familiarize yourself with the Java documentation: <https://docs.oracle.com/javase/8/docs/api/>

The most important part of the documentation is the API documentation. Find the documentation for the class `Scanner`. Read the examples at the top and have a look at which kind of elements can be read by the different `nextXXX()` methods.

- **Obligatory:** Modify `HelloWorld.java` using the topmost example from the `Scanner` documentation and the method `nextLine()` to prompt the user for a name and then greet the user by saying `Hello Mette!` if the user entered `Mette`. Continue with Exercise 2.3 from Chapter 2.
- *Supplementary:* Exercises 2.1–2.2 and 2.4–2.6.
- Challenge: Implement an Android app offering the all functionalities of Exercises 2.1–2.6. For example, it should ask you for a number of eggs and tell you how many grosses and dozens you have.