GHENT UNIVERSITY

Course Specifications

Valid as from the academic year 2025-2026

Programming (C003770)

Course size *(nominal values; actual values may depend on programme)*

Credits 6.0 **Study time 180 h**

Course offerings and teaching methods in academic year 2025-2026

A (semester 1)	Dutch	Gent	seminar
			lecture

Lecturers in academic year 2025-2026

Coolsaet, Kris	WE02	lecturer-in-charge

Offered in the following programmes in 2025-2026	**crdts**	**offering**
Bachelor of Science in Computer Science	6	A
Bachelor of Science in Mathematics	6	A

Teaching languages

Dutch

Keywords

Java, programming, object-oriented

Position of the course

[Please refer to the dutch version of this document for the most up to date information. This course cannot be taken by students that do not speak Dutch. Direct translation below is by ChatGPT]
Understanding and applying the basic principles of object-oriented programming using the Java programming language. Gaining practical programming experience in this domain and being able to write programs independently. Being able to evaluate whether code is of good quality, and improving its quality when necessary. Recognizing the importance (and actively acting on it) of the fact that a programmer or software developer rarely develops an entire program from scratch anymore; instead, programming primarily involves reusing existing program modules and integrating small pieces of code into existing software frameworks.

Contents

[Please refer to the dutch version of this document for the most up to date information. Direct translation below is by ChatGPT]

- Core concepts of object-oriented programming, including: objects, classes and interfaces, polymorphism, dynamic binding, inheritance, fields, constructors, and methods.
- Designing simple classes and interfaces (which attributes and methods are needed), dividing programs into classes, using classes for which only the specification is known, and implementing interfaces based on a specification.
- Writing clear and well-structured code ("programming with style"): choosing meaningful names, proper indentation, comments, and breaking code into methods.
- Handling runtime errors correctly using exceptions.
- Breaking down an assignment into subtasks to make the problem more manageable.
- Writing simple algorithms that use loops and basic data structures such as arrays and lists.
- Core principles for writing high-quality software with a focus on extensibility and readability. Concepts such as responsibility, encapsulation, coupling, cohesion, and refactoring. The concepts of mutators and accessors. Designing a suitable

class structure for a given problem.
- Applying these concepts and principles in the Java programming language, with additional focus on Java-specific elements, including:
  - Primitive data types and basic operations
  - One- and multi-dimensional arrays, and collection classes
  - Enumerated types (enums)
  - Controlling access to methods and fields: private vs. public (vs. protected)
  - Lambdas and streams
  - Checked and unchecked exceptions
  - Sequential text input and output

Initial competences

The student should know how to use a computer, but no prior programming experience is required.

Final competences

1 Analysing a task description in natural languages, and dividing it in subtasks.
2 Have practical and theoretical knowledge of the basic principles of object oriented programming (objects, classes, inheritance, interface vs. implementation, polymorphism, etc.)
3 Know how to convert a problem set in a natural language into an object oriented program, or part of a program, written in Java. For simple problems that only require basic library functions, the student should manage this without the use of a computer or documentation.
4 The student should be able to make good choices between alternative implementations bases on the quality criteria as studied in the course.
5 Know how to write a program unit which satisfies given specifications.
6 Know how to write a separate program unit as part of a bigger application largely written by others and for which only the specification and documentation is available (but no source code).
7 Know how to debug and test (part of) a program.

Conditions for credit contract

Access to this course unit via a credit contract is determined after successful competences assessment

Conditions for exam contract

This course unit cannot be taken via an exam contract

Teaching methods

Seminar, Lecture

Extra information on the teaching methods

[Please refer to the dutch version of this document for the most up to date information.]
Lectures and supervised computer labs. Newsgroup and web page facilities of the electronic class environment. The course book lends it self extremely well to (complementary) self-study.

Study material

Type: Handbook

Name: Programmeren in Java met BlueJ
Indicative price: € 70
Optional: no
Language : Dutch
Author : Same author
ISBN : 978-9-04303-499-9
Number of Pages : 664
Oldest Usable Edition : See Dutch version
Online Available : No
Usability and Lifetime within the Course Unit : regularly
Additional information: See Dutch version

Type: Other

Name: Bijkomende nota's
Indicative price: Free or paid by faculty
Optional: no

Language : Dutch
Author : de lesgever
Number of Pages : 0
Online Available : Yes

References

Java(TM) Language Specification (Java SE 24 Edition) James Gosling, Bill Joy, Guy Steele, Gilad Bracha, Alex Buckley. Also available online.

Course content-related study coaching

[Please refer to the dutch version of this document for the most up to date information.]
Continuous assistant supervision during lab classes, possibility to ask questions to the lecturer, personally or by e-mail. Newsgroup and web page facilities of the electronic classroom environment.

Assessment moments

end-of-term assessment

Examination methods in case of periodic assessment during the first examination period

Written assessment with open-ended questions

Examination methods in case of periodic assessment during the second examination period

Written assessment with open-ended questions

Examination methods in case of permanent assessment

Possibilities of retake in case of permanent assessment

not applicable

Calculation of the examination mark

100% exam