

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 2024-2025

A (semester 1)	Dutch	Gent	lecture seminar
----------------	-------	------	--------------------

Lecturers in academic year 2024-2025

Coolsaet, Kris	WE02	lecturer-in-charge
----------------	------	--------------------

Offered in the following programmes in 2024-2025

	crdts	offering
Bachelor of Arts in Moral Sciences	6	A
Bachelor of Arts in Philosophy	6	A
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.]

To know and apply the basic principles of object oriented programming from an 'objects first' perspective by means of the programming language Java. Obtain practical experience in this domain and know how to write programs independently. Realize the importance (and act accordingly) of the fact that a programmer or software developer seldom develops programs on his own, and that programming consists first and foremost of reusing existing modules and plugging smaller program units into larger existing software frameworks.

Contents

[Please refer to the dutch version of this document for the most up to date information.]

- Writing basic algorithms using recursion and loops
- Dividing a task into subtask to make them manageable
- Core concepts of object-oriented programming: objects, classes and interfaces, polymorphism, dynamic binding, inheritance, methods, fields, and constructors.
- Designing basic classes and interfaces (which methods and fields are needed?), decomposing a program into classes, using classes of which only the specification is known, implementing classes based on a specification.
- Core concepts for writing high quality software with a focus on extensibility and readability: responsibilities, encapsulation, coupling, cohesion, refactoring, mutator, accessor. Designing a good class hierarchy to solve a problem.
- Writing clean code: using proper names, indentation, comments, and splitting up long methods.
- Objects with reference and value semantics
- Reading and drawing class diagrams and object diagrams in UML.
- Properly handling errors during the execution of a program by using exceptions.
- Methods for testing and debugging programs.
- Applying these concepts and principles in the programming language Java. A number of Java specific elements will be studied in more detail.
 - Primitive types and operators

- 1-dimensional and multi-dimensional arrays and collection classes
- Accessibility modifiers (private, protected, public, package accessibility)
- Checked and unchecked exceptions

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 17 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