

## Object Oriented Programming (E702050)

Due to Covid 19, the education and evaluation methods may vary from the information displayed in the schedules and course details. Any changes will be communicated on Ufora.

Course size (nominal values; actual values may depend on programme)  
Credits 6.0 Study time 180 h Contact hrs 60.0 h

### Course offerings and teaching methods in academic year 2020-2021

A (semester 1)	Dutch	Gent	teaching methods	hours
			seminar: practical PC room classes	18.0 h
			online project	0.0 h
			lecture	18.0 h
			online lecture	6.0 h
			online seminar: practical PC room classes	18.0 h

### Lecturers in academic year 2020-2021

Naessens, Helga TW05 lecturer-in-charge

### Offered in the following programmes in 2020-2021

programme	crdts	offering
<a href="#">Bachelor of Science in Engineering Technology (main subject Electronics and ICT Engineering Technology)</a>	6	A
<a href="#">Bachelor of Science in Engineering Technology (main subject Information Engineering Technology)</a>	6	A
<a href="#">Linking Course Master of Science in Information Engineering Technology</a>	6	A

### Teaching languages

Dutch

### Keywords

Object oriented programming, Java, Computer Science (P170), Informatics (P175), Computer Technology (T120)

### Position of the course

This course learns the student the principles of object oriented programming in Java. Furthermore, this course has a broad educational value: it gives insight into abstract structures and processes, it develops analytical skills, the students learn to think modularly, they learn to solve problems themselves and to formulate appropriate solutions. The acquired theoretical knowledge and skills are used in many other areas (design, planning, optimization, ...)

This course is a fundamental course for other courses in the domain of computer science.

### Contents

Among other things following topics are covered:

- Basic principles of object oriented programming: classes, objects, methods, constructors, inheritance, overloading, Object, polymorphism, dynamic binding, interfaces
- Classes: organization and access, static, final, abstract
- Use of reference objects: arrays and shared structure, parameters and return values: privacy leaks, copy constructor, clone
- Exception handling
- IO
- Basic algorithms: searching, manipulating arrays, ...
- Collections, generics, iterators
- Internationalization
- Enumerations

- Lambda expressions and functional interfaces
- Streams

#### Initial competences

A good experience with some programming language (like for example Python): methods, sequence, selection, iteration, collections, ...

#### Final competences

- 1 Analyze, structure and translate a problem into a computer program in Java.
- 2 Apply the basic concepts of object oriented programming in Java (classes, object, methods, constructors, inheritance, overriding, overloading, Object, interfaces, polymorphism, dynamic binding, static, final, abstract, clone, collections, exception handling, IO, generics, iterators, internationalization, enumerations, lambda expressions and functional interfaces, streams, ...).
- 3 Independently implement, test and execute a computer program in Java.
- 4 Convert an object oriented design to a working computer program in Java

#### 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

Lecture, seminar: practical PC room classes, online lecture, online project, online seminar: practical PC room classes

#### Extra information on the teaching methods

During the lectures (24 h) the theory is explained step by step, partly based on examples.

During the exercise sessions (36 h) the student works independently on a PC.

#### Learning materials and price

Syllabus (Dutch) "Objectgeoriënteerd programmeren", sold by student organisation (estimated cost: 5 euro)

Slides, examples and exercises with solutions are provided on the electronic learning environment.

Some books about the course topics are available in the library.

#### References

Java How to Program, Early Objects, 11th Global Edition, Harvey Deitel & Paul J. Deitel, ISBN 978-1-2922-2385-8

Introduction to Java Programming and Data Structures, Comprehensive Version, Global Edition, 11/E, Y. Daniel Liang, ISBN 978-1-2922-2187-8

Core Java SE 9 for the Impatient, Cay S. Horstmann, ISBN 978-0-13-469472-6

#### Course content-related study coaching

The student can always make an appointment with the teacher.

#### Evaluation methods

end-of-term evaluation and continuous assessment

#### Examination methods in case of periodic evaluation during the first examination period

Written examination with open questions

#### Examination methods in case of periodic evaluation during the second examination period

Written examination with open questions

#### Examination methods in case of permanent evaluation

Assignment, skills test

#### Possibilities of retake in case of permanent evaluation

examination during the second examination period is possible in modified form

#### Extra information on the examination methods

The exam is a practical exam, consisting mainly of exercises, possibly complemented by a few theoretical questions.

For the lab section, the students have to make a project and there are skills tests on PC. If you don't pass the NPE, you will complete one test on all lab assignments in the second exam period. These points replace the quotation for the NPE.

#### Calculation of the examination mark

PE (written exam): 50%

NPE (skills test(s)): 50%

If the student obtains at least 8/20 for both parts, the score is calculated as the average of both parts (weights: 50 % exam, 50 % skills test(s)).

However, if the student does not obtain 8/20 or more for one or two parts, we deviate from this rule if the calculated final score is 10 or more. The score for this course will then be capped to 9/20.