

## Software Engineering (E017610)

**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 2023-2024**

A (semester 2)

Dutch

Gent

lecture

seminar

**Lecturers in academic year 2023-2024**

Dhoedt, Bart

TW05

lecturer-in-charge

**Offered in the following programmes in 2023-2024**

**crdts**

**offering**

[Bachelor of Science in Engineering\(main subject Computer Science Engineering\)](#)

6

A

[Preparatory Course Master of Science in Bioinformatics\(main subject Engineering\)](#)

6

A

**Teaching languages**

Dutch

**Keywords**

software engineering, software design, advanced JAVA

**Position of the course**

This course builds on basic programming knowledge and a basic understanding of algorithms and datastructures, and aims to teach methodologies to develop maintainable applications as well to introduce advanced programming concepts through the Java programming language.

**Contents**

- Methodologies for software development: Graphical software representation (UML), design patterns, software testing
- Advanced programming concepts: Frameworks and event driven programming, Graphical user interfaces, Error handling and exceptions, Polymorphism, interfaces and inner classes, input and output, serialisation, Runtime type identification, metaprogramming and reflection, Components, Multithreading, Generic types and collections

**Initial competences**

Basic knowledge on programming, algorithms and datastructures, e.g. as acquired through succesful completions of the courses "Informatics", "Discrete Mathematics" and "Programming"

**Final competences**

- 1 To build an architecture and a design for an application of limited complexity, using an UML-based methodology.
- 2 To know design patterns and apply them where appropriate.
- 3 To master the programming primitives related to multi-threading and to realise a multi-threaded program in Java.
- 4 To master object serialisation (including socket programming).
- 5 To know the concepts of software components and to master the Java Beans component model (including dynamic class loading and runtime type identification).
- 6 To realize a Java program, according to UML specification (including a graphical user interface).
- 7 To design unit tests, and to realise them in the JUnit framework.

**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

**Learning materials and price**

syllabus (20 Euro)

**References**

- Hans van Vliet, Software Engineering: Principles and Practice, John Wiley & Sons, 2000
- Martin Fowler, UML Distilled: A Brief Guide to the Standard Object Modeling Language
- Bruce Eckel, Thinking in Java
- Erich Gamma, Design Patterns

**Course content-related study coaching**

- interactive support through electronic learning environment
- guided exercises

**Assessment moments**

end-of-term and continuous assessment

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

Written assessment

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

Written assessment

**Examination methods in case of permanent assessment**

Skills test, Assignment

**Possibilities of retake in case of permanent assessment**

examination during the second examination period is not possible

**Extra information on the examination methods**

During examination period: written closed-book exam, additional exercise exam using PC

During semester: graded exercises. Second chance: Not possible

**Calculation of the examination mark**

Non-periodic and periodic evaluation. 10% of the score is on the lab exercises, 90% on the exam.