GHENT UNIVERSITY

Course Specifications

From the academic year 2018-2019 up to and including the

Software Engineering I (E761022)

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 2018-2019

A (semester 2)	Dutch	lecture	30.0 h
		practicum	30.0 h

Lecturers in academic year 2018-2019

Naessens, Helga	TW05	lecturer-in-charge
Van Den Breen, Wim	TW05	co-lecturer

Offered in the following programmes in 2018-2019	crdts	offering
Bachelor of Science in Engineering Technology (main subject Electromechanical Engineering Technology)	6	A
Bachelor of Science in Engineering Technology (main subject Information Engineering Technology)	6	A
Bachelor of Science in Electromechanical Engineering Technology	6	A
Bachelor of Science in Information Engineering Technology	6	A

Teaching languages

Dutch

Keywords

Systems analysis, Modelling, UML, Object oriented programming, object oriented design, GUI programming, JavaFX, Computer Science (P170), IT (P175), Computer Technology (T120)

Position of the course

In the first part of this course, the software development is treated. The aim is to enable the student to bring small projects to a successful conclusion. Methods are thaught to produce high-quality software.
In the second part of this course advanced topics of object-oriented programming are discussed. OGP aims at reuse and portability of code, but it requires a good understanding of the OGP concepts such as composition, inheritance, abstraction and interfaces.
The students also learn to program in JavaFX, enabling them to add a useful user interface to a program.

Contents

Part 1: Systems Analysis and Design

- Basics of good programming practice: characteristisation of good software and a good development.
- Reuse: how to reuse existing software and write code that can be reused.
- The different phases of the development process.
- Basics of UML.
- Requirements analysis and modelling: how to define the system to be developed.
- Design and realisation: converting a formal model to code.

Part 2: Java(FX) and API

- Advanced Topics in object oriented Java programming:
  - inheritance multiple classes:overriding, overloading, Object, polymorphism, dynamic binding
  - classes: organization and access
  - static, final, abstract
  - use of refernce objects: arrays and shared structure, copy constructor, clone, parameters and return values: privacy leak

    - exception handling
    - collections, generics, iterators
    - internationalization
    - enumerations
    - Git
    - unit testing
- GUI Programming in JavaFX: layout, events, graphics editing, controls and windows.

Initial competences

The competences acquired in Informatics II.

Final competences

1 Being able to apply principles of software design to the practice of production, maintenance and quality.
2 Being able to analyse, structure and translate a relatively complex problem into an object oriented design.
3 Being able to convert an object oriented design to a working computer program in Java and to test this program critically.
4 Being able to analyse, structure and translate a GUI problem into an effective GUI design.
5 Being able to turn this GUI design into a program working properly.

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, practicum

Extra information on the teaching methods

During the lectures the theory is explained step by step, partly based on examples. During the exercise sessions (attendance required) the student works independently or in pairs on a PC.

Learning materials and price

Slides, examples and exercises are provided on the electronic learning environment.
Syllabus (Dutch): ‘SOFTWAREONTWIKKELING I Partim: Geavanceerde topics OGP in Java' - 125 pp. (estimated cost max. 4 euro)
Syllabus (Dutch): ‘SOFTWAREONTWIKKELING I Partim: Systeemanalyse en -ontwerp ' - 60 pp. (estimated cost max. 2 euro)

References

- Ken Lunn: Software engineering met UML, Academic Service, 2004. ISBN 90 395 2253 7.
- Ivan Marsic: Software Engineering, Rutgers, 2009. Open Access Publishing. Downloadbaar op http://www.ece.rutgers.edu/~marsic/books/SE
- Perdita Stevens en Rob Pooley: Using UML. Software engineering with objects and components, Addison Wesley, 2000. ISBN 0 201 64860 1.
- Ken Arnold, James Gosling, David Holmes: The Java Programming Language, fourth edition
- Walter Savitch: Absolute Java, fifth edition
- David Flanagan: Java in a nutshell
- JavaFX Getting Started Tutorial: http://docs.oracle.com/javafx/2/get_started/jfxpub-get_started.htm
- JavaFX Documentation: http://docs.oracle.com/javafx
- JavaFX API: http://docs.oracle.com/javafx/2/api

Course content-related study coaching

Additional explanation by appointment or by e-mail.

Evaluation methods

end-of-term evaluation and continuous assessment

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

Written examination, oral examination

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

Written examination, oral examination

Examination methods in case of permanent evaluation

Participation, skills test

Possibilities of retake in case of permanent evaluation

examination during the second examination period is possible

Extra information on the examination methods

Theory: oral examination
Laboratory: continuous assessment, tests

Calculation of the examination mark

Theory: 50%
Laboratory: 50%
A weighted average is used to compute the final score for a training item. However, if a student gains a score of 7 or less on 20 on one of the different parts of this course, there will be deviated from the calculated final score if it is 10 or more and the score of the student will be a 9/20.
The tests of the labs can be retaken (possibly in modified form).