Course Specifications
Valid in the academic year 2019-2020

Software Engineering II (E761027)

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 2019-2020

- A (semester 1) Dutch seminar: practical PC room classes 36.0 h
- lecture 24.0 h

Lecturers in academic year 2019-2020

- Denert, Marleen TW05 staff member
- Maenhaut, Pieter-Jan TW05 staff member
- Ongenae, Veerle TW05 lecturer-in-charge

Offered in the following programmes in 2019-2020

<table>
<thead>
<tr>
<th>Programme</th>
<th>crdts</th>
<th>offering</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bachelor of Science in Engineering Technology (main subject Information Engineering Technology)</td>
<td>6</td>
<td>A</td>
</tr>
<tr>
<td>Linking Course Master of Science in Information Engineering Technology</td>
<td>6</td>
<td>A</td>
</tr>
</tbody>
</table>

Teaching languages

Dutch

Keywords

- C#, Design Patterns, Dependency Injection, AOP, Computer science (P170), Informatics (P175), Computer technology (T120)

Position of the course

The purpose of this course is to teach students advanced object oriented programming and design: specialized topics like concurrency, lambda expressions, delegates, ... , design patterns and the basic principles of dependency injection and aspect oriented programming.

Contents

- C#: introduction, threads, delegates, events, lambda expressions, GUI, ...
- Introduction to "Dependency Injection".
- Introduction to "Aspect Oriented Programming".
- Testing, debugging and logging
- Concurrency

Initial competences

- Being able to program and design in an object oriented way on an advanced level.
- Basic knowledge about GUI programming.

Final competences

1. Being able to develop a GUI program using threads, delegates, events, lambda expressions, ...
2. To have insight into the available "design patterns " for software design and for typical software problems.
3. To be able to know when which pattern is useful.
4. Being able to develop programs using patterns in a suitable way.
5. To be able to refactor programs according to some patterns.

(Approved)
6 Knowing the basics principles of dependency injection and being able to apply them.
7 Knowing the basics principles of Aspect Oriented Programming and be able to implement it.

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

Extra information on the teaching methods
- Lectures (24 hrs)
- Labs (36 hrs): individual work on PC; presence required.

Learning materials and price
"C# 3.0 Design Patterns", Judith Bishop, O'Reilly, 2008, completed with teacher's course (Dutch), slides and examples of programming.

References
“Head First Design Patterns”, Eric Freeman, Elisabeth Robson, Bert Bates & Kathy Sierra, O'Reilly Media

“Design Patterns: Elements of Reusable Object-Oriented Software”, Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, Addison-Wesley

Course content-related study coaching
The student can always make an appointment with the teachers.

Evaluation methods
end-of-term evaluation and continuous assessment

Examination methods in case of periodic evaluation during the first examination period
Written examination, skills test

Examination methods in case of periodic evaluation during the second examination period
Written examination, skills test

Examination methods in case of permanent evaluation
Skills test

Possibilities of retake in case of permanent evaluation
examination during the second examination period is not possible

Extra information on the examination methods
Several computer tests on PC and same tasks during the labs.

Calculation of the examination mark
Exam: 60% (written examination (60%) and computer exercises(40%))
Exercises/Labs: 40% (tests en tasks)

In the second examination period: score = maximum (E, 40% L + 60% E), where L is the score of the lab and E the score of the exam in the second examination period

(Approved)