GHENT UNIVERSITY

# Course Specifications

Valid in the academic year 2023-2024

## Algorithms and Data Structures (E018310)

**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**

| | | |
|---|---|---|
| Dhaene, Tom | TWO5 | lecturer-in-charge |
| Develder, Chris | TWO5 | co-lecturer |

| 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**

English, Dutch

**Keywords**

Algorithms, data structures

**Position of the course**

This course builds on earlier acquired basic programming skills, and deepens the knowledge and understanding of algorithms and data structures. Emphasis is put on techniques for systematic problem solving using algorithms, analysis and design of algorithms, abstract data types, and software implementation. The students will apply this knowledge to analyze engineering applications and to develop efficient programs in Python and/or C++.

**Contents**

**Lectures and seminars:**

- Algorithms: techniques for complexity analysis, algorithmic strategies (e.g. exhaustive methods, recursive strategies, divide-and-conquer strategies), comparison of strategies:
  - *Intro:* Basic concepts of complexity of algorithms, application to sort/search algorithms
  - *Analysis of algorithms:* techniques for complexity analysis, O-notation, recurrences
  - *Amortised analysis*
- Data types: abstract data types, including implementation and complexity analysis
  - *Hash tables*
  - *Binary search trees, including red-black trees, B-trees, augmenting data structures*
  - *Heaps and Fibonacci heaps*
- Quicksort, sorting in linear time and selection
- Dynamic programming
- Greedy algorithms
- NP-complete

**PC room classes and project:**

- Implementation of basic ADTs in Python (queues, hash tables, etc.)
- Data structures in C++ (standard template library, STL)

**Initial competences**

Basic programming skills, and basic knowledge of algorithms and data structures (e.g. as acquired in the courses 'Discrete Mathematics' and 'Informatics'). Programming skills in Python and basic knowledge of C++ (e.g. as acquired in the course 'Programming' (studied before or in parallel with this course)

**Final competences**

1 Know the key data structures.
2 List the most important basic algorithms for engineering applications and understand the philosophy behind them.
3 Determine the complexity of algorithms.
4 Select the most appropriate data structures and algorithms for a given problem, and provide motivation for the chosen structure/algorithm.
5 Critically assess new and existing software implementations in terms of their scalability and complexity.
6 Apply the main Python/C++ libraries for algorithms and data structures.
7 Apply and implement data structures and algorithms, possibly using existing libraries.

**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, Independent work

**Extra information on the teaching methods**

- Lectures
- Seminar: coached exercises & practical PC room classes

**Learning materials and price**

- Book: Introduction to Algorithms, Fourth Edition, Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, EAN 9780262046305. Approx. €95.
- Syllabus
- All course material is available in English

**References**

**Course content-related study coaching**

Interactive support via the electronic learning platform, guidance of the project work in feedback sessions.

**Assessment moments**

end-of-term and continuous assessment

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

Oral assessment, Written assessment

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

Oral assessment, Written assessment

**Examination methods in case of permanent assessment**

Assignment

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

examination during the second examination period is possible in modified form

**Extra information on the examination methods**

- During examination period: written closed-book exam complemented with oral examination.
- During semester: graded tasks and project work

**Calculation of the examination mark**

- Evaluation during examination period: 80%.
- Evaluation throughout semester: 20%.

You need to obtain a score of at least 9/20 on each of both parts (NPE and PE) in order to obtain a credit. Students who do not fulfill this condition but for whom the calculated score would be 9/20 or more, will receive a score of 8/20 (i.e., the largest score that is smaller than 9/20).
For the second examination chance, the student shall not retake the non-periodical evaluation. The score for the second examination chance is the maximum of the score for the periodical evaluation and the score obtained following the same calculation method as for the first examination chance.