

## Algorithmic Programming (E018321)

**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 2024-2025**

Offering	Language	Location	Teaching Methods	Hours
A (semester 1)	English	Gent	seminar	30.0h
			lecture	30.0h
			independent work	0.0h
B (semester 1)	Dutch	Gent	independent work	0.0h
			seminar	0.0h

**Lecturers in academic year 2024-2025**

Leyman, Pieter	TW18	lecturer-in-charge
Aelterman, Jan	TW07	co-lecturer

**Offered in the following programmes in 2024-2025**

Programme	crdts	offering
<a href="#">Bridging Programme Master of Science in Industrial Engineering and Operations Research(main subject Manufacturing and Supply Chain Engineering)</a>	6	A
<a href="#">Bridging Programme Master of Science in Industrial Engineering and Operations Research(main subject Transport and Mobility Engineering)</a>	6	A
<a href="#">Master of Science in Industrial Engineering and Operations Research(main subject Manufacturing and Supply Chain Engineering)</a>	6	A
<a href="#">Master of Science in Industrial Engineering and Operations Research(main subject Sustainable Mobility Analytics)</a>	6	A
<a href="#">Master of Science in Industrial Engineering and Operations Research(main subject Transport and Mobility Engineering)</a>	6	A
<a href="#">Master of Science in Industrial Engineering and Operations Research</a>	6	B

**Teaching languages**

English, Dutch

**Keywords**

Algorithms, data structures, relational databases, complexity, programming.

**Position of the course**

In various IE/OR courses, algorithms are taught to analyze and optimize performance of production, logistics, transport, and mobility systems. This course focuses on understanding the complexity of such algorithms and on the selection of appropriate data structures for an efficient implementation. Hands-on programming exercises are included for basic data analysis, data querying as well as for some basic optimization algorithms.

**Contents**

Lectures:

- Complexity analysis of algorithms, O-notation.
- Algorithmic strategies (exhaustive, incremental, recursive, divide-and-conquer), applied to searching and sorting
- Abstract data types: queue, stack, priority queues, dictionary, graph
- Data structures: array, linked list, heap, binary search tree, hash table
- Overview of database models, the relational model, and SQL
- Dynamic programming
- Greedy algorithms and greedy heuristics

Exercises:

- Complexity analysis exercises.

- Basic data analytics and data visualization.
- Manipulating and retrieving data from relational databases using SQL.
- Algorithmic programming – implementation of elementary OR algorithms:
  - Branch-and-bound for knapsack problem
  - Dijkstra's algorithm for shortest path
  - Prim/Kruskal's algorithm for minimum spanning tree
  - Greedy construction and improvement heuristic for routing or scheduling

### Initial competences

Basic knowledge of discrete mathematics, basic programming skills.

### Final competences

- 1 Determine algorithm complexity.
- 2 Distinguish database models and approach SQL-based relational databases from the user perspective.
- 3 Choose appropriate data structures for implementing an algorithm and motivate the choice.
- 4 Efficiently implement and apply basic data analysis and optimization algorithms.
- 5 Organize and document your workflow so others can reproduce, reuse, and extend your code.

### 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

### Study material

Type: Slides

Name: Slides and lecture notes

Indicative price: Free or paid by faculty

Optional: no

Language : English

Available on Ufora : Yes

Online Available : Yes

Available in the Library : No

Available through Student Association : No

### References

- Th. Cormen et al. Introduction to Algorithms (4th ed.). MIT Press, 2022.
- E.D. Taillard. Design of Heuristic Algorithms for Hard Optimization. Springer, 2023.
- Th. Mailund. Introduction to Computational Thinking. Apress, 2021.
- K.D. Lee and S. Hubbard. Data Structures and Algorithms with Python. Springer, 2015.

### Course content-related study coaching

#### Assessment moments

end-of-term and continuous assessment

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

Written assessment with open-ended questions

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

Written assessment with open-ended questions

#### Examination methods in case of permanent assessment

Oral 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

- Permanent evaluation: students report orally on several programming assignments.
- Periodic evaluation: written exam with open questions.

#### Calculation of the examination mark

Final score = 75% score exam + 25% score assignments

