

## Computability and Complexity (C000627)

**Course size** *(nominal values; actual values may depend on programme)*

**Credits 6.0**

**Study time 165 h**

**Course offerings and teaching methods in academic year 2024-2025**

A (semester 1)

English

Gent

seminar

lecture

**Lecturers in academic year 2024-2025**

Solda, Giovanni

WE16

lecturer-in-charge

Pakhomov, Fedor

WE16

co-lecturer

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

**crdts**

**offering**

[Master of Science in Teaching in Science and Technology\(main subject Mathematics\)](#)

6

A

[Master of Science in Computer Science](#)

6

A

[Master of Science in Mathematics](#)

6

A

[Exchange Programme in Computer Science \(master's level\)](#)

6

A

**Teaching languages**

English

**Keywords**

Complexity, computability, Turing machine, NP-completeness.

**Position of the course**

Theoretical computer science.

Be able to judge computability and complexity of problems, understand different concepts of computability and the connection between mathematical proofs and computability.

**Contents**

- 1 Models of computation, Turing machines
- 2 Complexity classes, P and NP
- 3 Polynomial reducibility
- 4 NP-completeness, satisfiability
- 5 Undecidable problems, the halting problem

**Initial competences**

It is useful to have attended a basic course on Discrete Mathematics and a course on Formal Languages and Finite Automata (such as the course Automata, Computability and Complexity), or to have acquired the competencies from such courses another way.

**Final competences**

- 1 The student is familiar with different concepts of computability and complexity and can compare them.
- 2 He or she can judge the complexity of various problems.
- 3 He or she can apply this knowledge to new problems.

**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

## Study material

Type: Handouts

Name: Lecture notes'

Indicative price: Free or paid by faculty

Optional: no

## References

- Michael Sipser: Introduction to the Theory of Computation, Cengage Learning, 2013, 481 pages, ISBN 978-1-133-18781-3
- S. Arora, B. Barak: Computational complexity. A modern approach. Cambridge University Press, Cambridge, 2009. 579 pp. ISBN: 978-0-521-42426-4.
- R.I. Soare: Turing computability: Theory and applications, Springer, 2016, 263pp, ISBN:978-3-642-31933-4.
- S.Homer and A.L. Selman: Computability and complexity theory, Springer, 2001, 194 pp, ISBN:0-387-95055-9.

## Course content-related study coaching

Individually -- students can always contact the lecturers.

## 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

Participation, 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

The permanent evaluation is based on presentations by the student and his active participation during the lectures.

## Calculation of the examination mark

Permanent evaluation: mark between 0 and 10.

Periodic evaluation: mark between 0 and 5 for each of written and oral exam, out of 10 in total.

The final mark is the sum of these two.