

## Fundamentals of Programming Languages (C003241)

**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)	Dutch	Gent	lecture seminar
----------------	-------	------	--------------------

**Lecturers in academic year 2024-2025**

Scholliers, Christophe	WE02	lecturer-in-charge
------------------------	------	--------------------

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

	<b>crdts</b>	<b>offering</b>
<a href="#">Master of Science in Teaching in Science and Technology(main subject Computer Science)</a>	6	A
<a href="#">Master of Science in Teaching in Science and Technology(main subject Mathematics)</a>	6	A
<a href="#">Master of Science in Computer Science</a>	6	A
<a href="#">Master of Science in Computer Science Engineering</a>	6	A
<a href="#">Master of Science in Mathematics</a>	6	A

**Teaching languages**

Dutch

**Keywords**

programming languages, type systems, operational semantics, lambda calculus

**Position of the course**

This course introduces the student to the formal study of programming language and the mathematical techniques used for that purpose.

**Contents**

The courses covers in a formal manner

- syntax
- operational semantics
- type systems
- program analysis based on hoare logic
- the most important properties of syntax, semantics, type systems and related proof methods, including
  - type preservation
  - progress
  - Curry-Howard isomorphism
  - logical relations
  - ...

The study is performed on the basis of the simply typed lambda-calculus and various extensions, such as

- primitive types
- tuples
- records
- sum types
- polymorphism
- ...

Not all topics are covered every year.

**Initial competences**

Experience with the practical use of a programming language, e.g. by means of the Programming/Programming 1 course.

Knowledge of first order logic and basic mathematical proof techniques such as induction.

### Final competences

- 1 To formalize informal statements about programming languages.
- 2 To prove properties of programming languages.
- 3 To use the most important concepts of type systems and hoare logic creatively in new applications.
- 4 To appreciate the formal treatment of programming languages.
- 5 Give proofs in the COQ proof assistant

### 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

### Extra information on the teaching methods

alternating lectures and exercise sessions

### Study material

Type: Software

Name: Software'

Indicative price: Free or paid by faculty

Optional: no

Additional information: Software Foundations online course, Benjamin C. Pierce, free website, free

### References

- Hennessy, M. (1990). The semantics of programming languages. Wiley. <http://www.scss.tcd.ie/Matthew.Hennessy/slexternal/reading.php>
- Winskel, G. (1993). The formal semantics of programming languages. MIT Press.
- Carl Gunter. Semantics of Programming Languages. MIT Press, 1992. (ISBN 0-262-07143-6)
- Robert Harper. Practical Foundations for Programming Languages. Working draft, 2006. (online, as PDF)
- Shriram Krishnamurthi. Programming Languages: Application and Interpretation. (online, as PDF)
- Mitchell, John C.. Foundations for Programming Languages.
- John C. Reynolds. Theories of Programming Languages. Cambridge University Press, 1998. (ISBN 0-521-59414-6)
- Kenneth Slonneger and Barry L. Kurtz. Formal Syntax and Semantics of Programming Languages. Addison-Wesley.
- Robert D. Tennent (1991). Semantics of Programming Languages. Prentice-Hall.
- H. Nielson and F. Nielson (1993) Semantics with Applications. A formal Introduction. Wiley

### Course content-related study coaching

interactive support on the electronic teaching system Ufora (forums, e-mail), appointments for personal support

### Assessment moments

end-of-term and continuous assessment

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

Oral assessment

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

Oral 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

### Calculation of the examination mark

project assignments: 20%

oral exam: 80%

When a student obtains a grade less than 10/20 for either the theory or the project, the total end grade will be maximally the highest failing grade 9/20.