

Functional Programming (C003775)

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 2018-2019

A (semester 1)	Dutch	lecture	30.0 h
		seminar: coached	30.0 h
		exercises	

Lecturers in academic year 2018-2019

Scholliers, Christophe WE02 lecturer-in-charge

Offered in the following programmes in 2018-2019

Bachelor of Science in Computer Science	crdts	offering
	6	A

Teaching languages

Dutch

Keywords

programming languages, functional programming, higher-order functions, polymorphism, monads, type classes, lazy evaluation, monads, monad transformers, Haskell

Position of the course

The student immerses herself/himself in the functional programming paradigms.

At the end of the course, she/he should:

- be able to make use of a functional programming language in practice;
- master the common concepts, programming techniques and data structures of this paradigm;
- have gained insight in the commonalities between functional programming and object-oriented programming
- have gained insight in the underlying evaluation mechanisms of these languages.

Contents

Haskell in depth, aspects of other functional languages
 evaluation mechanisms: lazy evaluation
 type system: algebraic datatypes, polymorphism, type classes
 higher-order functions continuations, functors, monads
 data structures: immutable, infinite

Initial competences

Being able to program in a programming language, e.g., by having taken the Programming course.

Final competences

- 1 The student is able to write small and medium size programs in a functional programming language.
- 2 She/he is able to apply the common concepts and data structures in this language in practice.

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: coached exercises

Extra information on the teaching methods

Lectures, exercise sessions: supervised exercises, in computer lab

Learning materials and price

Slides

Website

Estimated total price: 0 EUR

References

- Programming in Haskell, Graham Hutton, University of Nottingham, Cambridge University Press, January 2007.
- Simon Thompson: Haskell: The Craft of Functional Programming, Second Edition, Addison-Wesley, 507 pages, paperback, 1999.
- Learn You a Haskell for Great Good!: A Beginner's Guide 1st Edition by Miran Lipovaca (Author)

Course content-related study coaching

Supervised exercise/lab sessions

Electronic learning environment

Evaluation methods

end-of-term evaluation and continuous assessment

Examination methods in case of periodic evaluation during the first examination period

Written examination with open questions

Examination methods in case of periodic evaluation during the second examination period

Written examination with open questions

Examination methods in case of permanent evaluation

Oral examination, assignment, skills test

Possibilities of retake in case of permanent evaluation

examination during the second examination period is possible

Extra information on the examination methods

Permanent evaluation: assignment, proficiency test, oral defence

Second exam chance in case of permanent evaluation: new assignment

Calculation of the examination mark

Permanent evaluation (50%) + periodic evaluation (50%).

Students need to pass on both the theory and the practical assignments to pass i.e. when scoring less than 10/20 on one of the subjects the lowest score counts.