MI008	**Semantics of programming languages**	L	P	S	ECTS
		2	2	0	6

Course objectives. Introduce students with the mathematical analysis of programming languages. Students will study programming language concepts using the framework of typed lambda calculus. The main goal is to present programming language concepts and features beyond the surface syntax and to understand the meaning of program phrases (expressions, commands, declarations, etc.). All concepts will be related to functional programming paradigm and Haskell programming language.

Prerequisites. Undergraduate university study programme of mathematics and/or computer science.

Course content.

1. Introduction. Lambda notation. Axiomatic, operational, denotational semantics.
2. The language PCF (Programming Computable Functions). Syntax. Booleans and natural numbers. Pairing and Functions. Declarations and syntactic sugar. Recursions and Fixed-point operator. PCF Programs and their semantics. PCF reduction and symbolic interpreters. PCF Programming examples, expressive power and limitations. Variations and extension of PCF.
3. Universal algebra and algebraic data types. Algebras, signatures and terms. Equations, soundness and completeness. Homomorphism and initiality. Algebraic data types. Rewrite systems.
4. Simply-typed lambda calculus. Types. Terms. Proof systems. Henkin modles, soundness and completeness.
5. Imperative programs. While programs. Operational semantics. Denotational semantics. Before-after assertions about While programs. Semantics of additional program constructs.

LEARNING OUTCOMES

No.	LEARNING OUTCOMES
1.	Understand lambda calculus.
2.	Understand syntax and semantics of typed lambda calculus.
3.	Understand difference between axiomatic, operational and denotational semantics.
4.	Demonstrate standard programming paradigms in the context of functional programming language.
5.	Understand the two different models in the context of typed lambda calculus; partially ordered structure, recursive function theory.

RELATING THE LEARNING OUTCOMES, ORGANIZATION OF THE EDUCATIONAL PROCESS AND ASSESSMENT OF THE LEARNING OUTCOMES

TEACHING ACTIVITY	ECTS	LEARNING OUTCOME **	STUDENT ACTIVITY*	EVALUATION METHOD	POINTS	
					min	max
Attending lectures and exercises	1	1-5	Lecture attendance, discussion, teams work, independent work on given tasks and short written exams	Attendance lists, tracking activities, closed form exercises	0	4
Homework assignments	1	1-5	Independent work on given problems	Evaluation	0	4
Written exam (Mid-terms)	2	1-5	Preparing for written exam	Evaluation	25	46

Final exam	2	1-5	Revision	Oral or written exam	25	46
TOTAL	6				50	100

Teaching methods and student assessment. Lectures and exercises are obligatory. The exam consists of a written and an oral part. Upon completion of the course, students can take the exam. Successful midterm exam scores replace the written exam. Exercises are both auditory and laboratory. Laboratory exercises include the usage of computers. Students can improve their grades by writing homework assignments and seminars.

Can the course be taught in English: Yes

Basic literature:

1. J.C. Mitchell, Foundations for Programming Languages, MIT Press, 1996.

Recommended literature:

1. J.R. Hindley, J.P. Seldin, Lambda-Calculus and Combinators, an Introduction, Cambridge University Press 2008.
2. F. Baader, T. Nipkow, Term Rewriting and All That, Cambridge University Press 1998.
3. M. R. A. Huth, M.D. Ryan, Logic in Computer Science, Modelling and Reasoning about Systems, Cambridge University Press, 2000.