Study programmes: PhD studies - Informatics

Course name: P442 - Semantics of programming languages

Lecturers: Milena Vujošević Janičić, Filip Marić and other lecturers of the Department for Computer Science

Status: Optional

ECTS: 9

Attendance prerequisites: No prerequisites

Course aims: Acquiring knowledge about different programming language semantics, i.e. about operational, denotational and axiomatic semantics, understanding their relationship, advantages and disadvantages, and possible applications in analysis, optimization and verification of programs.

Course outcome: After the course, students have adopted the most important static and dynamic approaches of software verification and gained practical experience with different verification tools.

Course content:

- Operational semantics and implementation of programing languages: natural semantics, structural operational semantics, abstract machines, provably correct implementation
- Denotational semantics and software analysis: denotational semantics, fixed point theory, analysis and transformation of programs
- Axiomatic semantics and software verification: axiomatic semantics, specification and software verification, weakest precondition, generating verification conditions

Literature:

1. Nielson and Nielson: Semantics with Applications: An Appetizer, Springer-Verlag, 2007, ISBN: 978-1-84628-691-9.
2. Glynn Winskel: The Formal Semantics of Programming Languages. MIT Press, 1993, ISBN:0-262-23169-7.
3. Tobias Nipkow, Gerwin Klein: Concrete Semantics. A Proof Assistant Approach, 2014.
(the lecturer can choose another appropriate literature)

Number of hours: 10	**Lectures: 4**	**Tutorials: -**	**Laboratory: -**	**Research: 6**

Teaching and learning methods: Frontal/Individual/Group work/Practical work.

Assessment (maximal 100 points)

Course assignments	**points**	**Final exam**	**points**
Lectures	10	Written exam	-
Exercises / Tutorials	-	Oral exam	-
Colloquia	20	Written-oral exam	50
Essay / Project	20		