M097	**Theoretical Foundations of Computer Science**	L	S	E	ECTS
		2	0	2	6

Course objectives. Course objectives are to familiarize students with abstract machines and problems they are able to solve (automata theory) and formal languages (regular languages and context-free languages). Emphasis will be put on students' way of thinking that enables better understanding of this field. During exercises students will get to know many examples of formal languages in different programming languages.

Course prerequisites. Introduction to Computer Science. Mathematical logic in Computer Science

Syllabus.

1. Automata. Computability. Complexity.
2. Regular languages: Finite Automata. Nondeterminism. Regular Expressions. Non-regular Languages.
3. Context-Free Languages: Context-Free Grammars. Pushdown Automata. Non-Context-Free Languages. Deterministic Context-Free Languages.
4. Decision problems for Context-Free Languages. Turing machines.
5. Decidability.
6. Rice's Theorem
7. Time complexity. P and NP. NP-completeness.
8. Polynomial time reductions (NP-complete reductions).

EXPECTED LEARNING OUTCOMES

No.	LEARNING OUTCOMES
1.	To demonstrate knowledge and understanding which can serve as the foundation for developing and application of original ideas.
2.	To apply knowledge, understanding and skills in a broad variety of problems in the field of automata theory and formal languages.
3.	To integrate new knowledge and problem solving skills in this field.
4.	To be able to present conclusions and findings to experts and laymen based on knowledge and experience.
5.	To apply the acquired skills onto further education in this field.

COUPLING OF THE EXPECTED LEARNING OUTCOMES, TEACHING PROCESS ORGANIZATION AND THE EVALUATION OF THE TEACHING OUTCOMES

TEACHING PROCESS ORGANIZATION	ECTS	EXPECTED LEARNING OUTCOMES **	STUDENT ACTIVITY *	EVALUATION METHOD	SCORE	
					min	max
Lecture attendance	1	1-5	Class attendance, discussion, solving the	Lists with signatures, observing the	0	4

			problems individually and in a team	activity during the lectures		
Homework	1	1-4	Solving the problems individually	Grading	12	20
Repeated exams	2	1-4	Preparation for the written exam	Grading	19	38
Final exam	2	1-4	Revising	Oral exam	19	38

Teaching methods and student assessment. Classes are organized through lectures and exercises. During lectures students will be familiarized with basic terms and results in mathematical theory of computation through illustrative examples and/or proofs. Exercises are auditory. During exercises students apply the acquired abstract knowledge to concrete problems. Lectures and exercises are obligatory. Final examination consists of a written and oral part which is taken after the completion of lectures. Acceptable mid-term exam scores replace the written examination. Homework and seminar papers done during the semester could influence the final result of the exam.

Can the course be taught in English: Yes

Basic literature:

1. M. Sipser, Introduction to the Theory of Computation (3rd Ed.), Cengage Learning, Boston, 2013.

Additional literature:

1. S. Arora, B. Barak, Computational Comlexity, A Modern Approach, Cambridge University Press, 2009.
2. J. Hromkovič, Theoretical computer science, Springer Verlag, 2004
3. J. E. Hopcroft, R. Motwani, J. D. Ullman, Introduction to Automata Theory, Languages and Computation (Third Edition), Addison Wesley, 2006.
4. A. Maruoka, Concise Guide to Computation Theory, Springer Verlag, London, 2011
5. O. Goldreich, Computational Complexity - A Conceptual Perspective, Cambridge University Press, 2008
6. C. H. Papadimitriou, Computational Complexity, Addison Wesley, 1993
7. H. Lewis, C. H. Papadimitriou, Elements of the Theory of Computation (Second Edition), Prentice Hall, 1997
8. Z. Manna, Mathematical Theory of computation, McGraw-Hill Inc. , 1974.