SYLLABUS

Basic information of the course	
University:	**University “Ukshin Hoti” - Prizren**
Academic unit:	**Faculty of Computer Science**
Study program:	**Information Technologies and Telecommunication**
Course:	**Finite Automata and Formal Languages**
Study level:	**Bachelor**
Course status:	**Elective**
Study year:	**3**
Number of hours per week:	**2+2**
Credit value - ECTS:	**6**
Time / location:	**It will be published in the university web site!**
Lecturers:	**Assoc. Prof. Dr. Samedin Krrabaj** **Ass. Arbër Beshiri, Ph. D. c.**
Contact details:	**samedin.krrabaj@uni-prizren.com** **arber.beshiri@uni-prizren.com**
Course description:	The course introduces some fundamental concepts in automata theory and formal languages including grammar, finite automaton, regular expression, formal language, pushdown automaton, and Turing machine. Not only do they form basic models of computation, they are also the foundation of many branches of computer science, e.g. compilers, software engineering, concurrent systems, etc. The properties of these models will be studied and various rigorous techniques for analyzing and comparing them will be discussed, by using both formalism and examples. It will be presented abstract models of computers (finite automata, push down automata and Turing machines) and the language classes they recognize or generate (regular, context-free and recursively enumerable). Also presents applications of these models to compiler design, algorithms and complexity theory.
Course objectives:	This course is developed to introduce students to: - Demonstrate an understanding of abstract models of computing, including deterministic

	(DFA), non-deterministic (NFA), and Turing (TM) machine models; - Understand the relative computing power of the different abstract machine models; - Demonstrate an understanding of regular expressions and grammars, including context-free and context-sensitive grammars; - Understand the relationships between language classes, including regular, context-free, context-sensitive, recursive, and recursively enumerable languages; - Understand the associations between language classes and machine models; - Understand the associations between language classes and language descriptors (i.e., grammars and regular expressions); - Understand what decidable and undecidable problems are; - Apply advanced proof techniques such as reductions and diagonalization (scientific, computing, and engineering problem solving); - Understand the application of machine models and descriptors to compiler theory and parsing.
Learning outcomes:	After completion of this course, students will be able to: - Identify different formal language classes and their relationships; - Learn and use structural induction; - Work with models of computing and understand their different powers; - Understand and formulate regular expressions; - Understand and formulate context-free grammars; - Be familiar with Turing machines and computability.

Contribution on student load (must correspond with learning outcomes)			
Activity	**Hours**	**Days/week**	**Total/hours**
Lectures	2	15	30
Exercise theoretical/laboratory	2	15	30
Practice work	0	0	0
Contact with	1	5	5

lecturer/consultations			
Field exercises	1	1	1
Midterms	2	2	4
Laboratory exercises	2	2	4
Individual time spent studying (at the library or home)	4	10	30
Final preparation for the exam	5	6	30
Time spent in evaluation (tests, quiz, final exam)	2	3	6
Projects, presentations, etc.	0	0	0
Total			**150**
Notice: 1 ECTS credits = 25 hours commitment, e.g. if the course has 6 ECTS credits student must have 150 hours during the semester.			

Teaching methods:	The course is a combination of lectures, discussions, numerical and laboratory exercises, while the assignments are presented by the laboratory course lecturers!
Assessment methods:	- Attendance in lectures and exercises: 5% + 5% = 10% - Laboratory assigments: 20%. - Midterm 1: 35%. - Midterm 2: 35%. - Or final exam: 100%.

Assessment and grading:

Vlerësimi në %	Nota përfundimtare
91% - 100%	10
81% - 90%	9
71% - 80%	8
61% - 70%	7
51% - 60%	6
0% - 50%	5

Literature	
Basic literature:	1. Peter Linz. An Introduction to Formal Languages and Automata, 6th Edition, Jones & Bartlett Learning, 2017.
Additional literature:	1. Michael Sipser. Introduction to the Theory of Computation, 3rd Edition, Cengage Learning, 2012. 2. John E. Hopcrof, Rajeev Motwani, Jeffrey D. Ullman. Introduction to Automata Theory, Languages, and Computation, 3rd Edition, Pearson/Addison-Wesley, 2007.

Study plan

Week	Lectures
First week:	• Introduction to course organization - syllabus (about lectures). • Inductive proofs.
Second week:	• Relations and functions.
Third week:	• Diagonalization.
Fourth week:	• Regular languages: DFA, NFA, and NFA-& epsilon machines.
Fifth week:	• Equivalence of DFA and NFA machines.
Sixth week:	• Equivalence of regular expressions and finite state machines.
Seventh week:	• Pumping lemma for regular languages.
Eighth week:	• First midterm.
Ninth week:	• Context-free languages, pushdown automata, pumping lemma for context-free languages, etc.
Tenth week:	• Recursive and recursively enumerable languages.
Eleventh week:	• Turing machines.
Twelfth week:	• Definition of recursive and recursively enumerable. • Methods for Turing machine construction.
Thirteenth week:	• Modifications of the basic Turing machine model • Equivalence of the different TM models and the basic TM model.
Fourteenth week:	• TMs as enumerators. • Decidability, undecidability. • The halting problem: undecidability of the halting problem
Fifteenth week:	• Second (final) midterm.

Exercises

Study plan	
Java	**Exercises**
First week:	• Introduction to course organization - syllabus (about lectures). • Inductive proofs.
Second week:	• Relations and functions.
Third week:	• Diagonalization.
Fourth week:	• Regular languages: DFA, NFA, and NFA-& epsilon machines.
Fifth week:	• Equivalence of DFA and NFA machines.
Sixth week:	• Equivalence of regular expressions and finite state machines.
Seventh week:	• Pumping lemma for regular languages.
Eighth week:	• First midterm.

Ninth week:	• Context-free languages, pushdown automata, pumping lemma for context-free languages, etc.
Tenth week:	• Recursive and recursively enumerable languages.
Eleventh week:	• Turing machines.
Twelfth week:	• Definition of recursive and recursively enumerable. • Methods for Turing machine construction.
Thirteenth week:	• Modifications of the basic Turing machine model • Equivalence of the different TM models and the basic TM model.
Fourteenth week:	• TMs as enumerators. • Decidability, undecidability. • The halting problem: undecidability of the halting problem
Fifteenth week:	• Second (final) midterm.

Academic policies and rules of conduct

- Generally lecture presentations will be made through MS PowerPoint, tables, material usage, computer programs and numeric exercises.
- Additional resources (scientific papers, publications, national bulletins, as well as recent discoveries and research) will be provided by professors.
- In the absence of the opportunity for practical work to be organized weekly, in cooperation with the management of the university, this activity will be organized on certain days in: organizations, companies, etc.
- During each session will be organized the conversation and co-participation with the students!
- Students are required to be regular in lectures and exercises!
- It will be evaluated when the students collaborate and participate in the lectures and course exercises!
- Timely arrival in lectures and exercises is mandatory!