Identification	**Subject**	CMS 370 –Introduction to Theory of Computation – 3KU /6ECTS credits
	Department	Computer Science
	Program	Undergraduate
	Term	Spring, 2017
	Instructor	Niyamaddin Taghiyev
	E-mail:	taghiyevniyam@gmail.com
	Phone:	055 311 68 00
	Classroom/hours	11 Mehseti str. (Neftchilar campus)
	Office hours	not available
Prerequisites		
Language	English	
Compulsory/Elective	Required	
Description	This course emphasizes theoretical models of computation and their analysis. The aim of analysis is to identify and prove the capabilities and limitations of particular models of computation. It is shown that there are problems that are unsolvable, that is, there are questions that cannot be answered by any model of computation. Limits on computation in the context of resource bounds are also investigated. Techniques (reductions) are developed to show that one model of computation is equivalent in power to another or that it is different in power from another. Models of computation that are covered include finite automata, pushdown automata, and Turing machines. Some complexity theory is covered as well.	
Required textbooks and course materials	***Core textbook:*** Introduction to the Theory of Computation (Third Edition). Michael Sipser. Thomson Course Technology, 2013. ISBN: 978-1-133-18779-0.	
Course objectives	Finite automata are useful models for many important kinds of hardware and software. Here are the most important kinds: Software for designing and checking the behavior of digital circuits; The “lexical analyzer” of a typical complier, that is, the compiler component that breaks the input text into logical units, such as identifiers, keywords, and punctuation; Software for scanning large bodies of text, such as collections of Web pages, to find occurrences of words, phrases, or other patterns; Software for verifying systems of all types that have a finite number of distinct states, such as communication protocols or protocols for secure exchange of information.	
Learning outcomes	Students will learn several formal mathematical models of computation along with their relationships with formal languages. In particular, they will learn regular languages and context free languages which are crucial to understand how compilers and programming languages are built. Also students will learn that not all problems are solvable by computers, and some problems do not admit efficient algorithms. Throughout this course, students will strengthen their rigorous mathematical reasoning skills. At the end of this course, students will be able to do the following: • Students will demonstrate knowledge of basic mathematical models of computation and describe how they relate to formal languages. • Students will understand that there are limitations on what computers can do, and learn examples of unsolvable problems. • Students will learn that certain problems do not admit efficient algorithms, and identify such problems.	

Teaching methods	**Lecture**		x
	Experiential exercise		x
	Assisted work		x
	Assisted lab work		x
	Others		
Evaluation	**Methods**	**Date/deadlines**	**Percentage (%)**
	Midterm Exam		25
	Class Participation and Attendance		10
	Quizzes		20
	Lab Exercises		
	Project and Assignment		10
	Final Exam		35
	Total		100

Policy

- **Preparation for class**

The structure of this course makes your individual study and preparation outside the class extremely important. The lecture material will focus on the major points introduced in the text. Reading the assigned chapters and having some familiarity with them before class will greatly assist your understanding of the lecture. After the lecture, you should study your notes and work relevant problems and cases from the end of the chapter and sample exam questions.
Throughout the semester we will also have a large number of review sessions. These review sessions will take place during the regularly scheduled class periods.

- **Project**

The objective of this assignment is to create Software Application using methods, algorithms and techniques taught in class. The assignment is organized on the basis that students should work on it in groups. Groups must be composed of a maximum 3 members. We strongly encourage the students to work in pair. The assignment is divided into two parts: Soft and Presentation.
Points will be deducted for assignments turned in after the deadline.

- **Withdrawal (pass/fail)**

This course strictly follows grading policy of the School of Engineering and Applied Science. Thus, a student is normally expected to achieve a mark of at least 60% to pass. In case of failure, he/she will be required to repeat the course the following term or year.

- **Cheating/plagiarism**

Plagiarism and Cheating of any kind on an examination, quiz, or project will lead to assignment cancellation. In this case, the student will automatically get zero (0), without any considerations.

- **Professional behavior guidelines**

The students shall behave in the way to create favorable academic and professional environment during the class hours. Unauthorized discussions and unethical behavior are strictly prohibited.

- **Ethics**

Students should not arrive in late to class. All electronic devices must be silenced and stowed during class.

Tentative Schedule			
Weeks	**Date/Day (tentative)**	**Topics**	**Textbook/Assignments**
1		**Review of Mathematical Theory:** Sets, Functions, Logical statements, Proofs, relations, languages, Mathematical induction, strong principle and Recursive definitions	
2		**Finite Automata and Regular Languages:** Deterministic and Nondeterministic Finite Automata.	
3		**Finite Automata and Regular Languages:** Equivalence of DFA and NFA, and regular expressions.	
4		**Finite Automata and Regular Languages:** Regular languages and Non-regular languages and pumping Lemma, and closure properties.	
5		**Context-Free Languages:** Context-free grammar and Chomsky Normal Form.	
6		**Context-Free Languages:** Pushdown automata and Non-Context-free languages.	
7		**Computability Theory:** Turing Machines and The Church-Turing Thesis	
8		**Midterm Exam** **Mid-term report**	
9		**Computability Theory:** Decidability and Rice's Theorem	
10		**Computability Theory:** Mapping Reductions, More Undecidable Languages	
11		**Complexity Theory:** Time Complexity, the class P and NP	
12		**Complexity Theory:** NP-Completeness	
13		**Complexity Theory:** Space Complexity	
14		**Complexity Theory:** The Classes L and NL and NL-completeness	
15		**Advanced Topics in Complexity Theory and Review**	

This syllabus is a guide for the course and any modifications to it will be announced in advance.