OSTIM TECHNICAL UNIVERSITY
FACULTY OF ENGINEERING
SOFTWARE ENGINEERING
UNDERGRADUATE COURSE

COURSE SYLLABUS FORM
2021-2022 FALL

YZL 305 Formal Languages and Automata							
Course Name	**Course Code**	**Term**	**Hour**	**Practice**	**Lab**	**Credit**	**ECTS**
Formal Languages and Automata	YZL 305	5	3	0	0	3	5

Language of the Course	English
Type of Course	Mandatory
Course Level	Undergraduate
Method of Teaching	Online
Course Learning and Teaching Techniques	Lecture, Q/A, Homework

Purpose of the Course

The aim of this course is to provide students with the necessary theoretical knowledge to design and analyze systems that perform discrete calculations.

Learning Outcomes

Students who successfully complete this course;

- Knowledge necessary to understand abstract machine models and formal languages,
- will acquire the ability to design abstract machine models that can accept various formal languages.

Course Content

The theory of mathematical models of computers constructed through abstract machines and their corresponding formal languages. Formal languages, grammars, finite state machines, regular sets, regular expressions, boundaries of finite state models, pushdown automata, context-free languages, Turing machines, efficient computability, unsolvable decision problems.

Weekly Plan and Related Preparation Studies	
Week	**Subjects**
1	Introduction, Proof Methods
2	Finite Automata
3	Regular Expressions
4	Features of Regular Languages
5	Decision Properties of Regular Languages
6	Context-Free Grammars and Ambiguity
7	Pushdown Automata
8	Midterm Exam
9	Equivalence of Pushdown Automata and Context-Free Grammars
10	Operations on Context-Free Grammars
11	Closeness Properties of Context-Free Grammars
12	Turing Machines and Complexity
13	Different Turing Machine Models
14	Decided and Undecided Problems
15	NP-Full Problems
16	Final Exam

Resources (Textbook and supplementary book)
1. Automata Theory, Languages and Computation, by John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman. (Pearson – 3rd Edition)

Evaluation System		
Studies	**Number**	**Contribution**
Attendance		
Lab		
Application		
Field Study		
Course Specific Internship (if applicable)		
Quizzes/Studio/Critical		
Homework		
Presentation		
Projects		
Report		
Seminar		
Midterm Exams/Midterm Jury	1	40%
General Exam/Final Jury	1	60%
	Total	**100%**
Contribution of Mid-Semester Studies to Success Grade		50%
Contribution of End of Semester Studies to Success Grade		50%
	Total	**100%**

Course Category	
Basic Vocational Courses	
Specialization/Field Courses	x
Support Lessons	

Communication and Management Skills Lessons	
Transferable Skills Lessons	

Course Learning Outcomes and Program Qualifications							
No	**Program Qualifications / Outcomes**	**Contribution Level**					
		1	**2**	**3**	**4**	**5**
1	Ability to apply mathematics, science and engineering				x	
2	Ability to design and conduct experiments and to analyze and interpret experimental results.					
3	Ability to design a system, component, and process and according to specified requirements.				x	
4	Ability to work in an interdisciplinary team.				x	
5	Ability to identify, formulate and apply engineering problems.					x
6	Identifies, defines, formulates, solves complex Software Engineering problems and chooses and applies analysis and modelling methods suitable for this purpose.				x	
7	Develops, selects, uses modern techniques and tools necessary for the analysis and solution of complex problems encountered in Software Engineering applications and uses information technologies effectively.				x	

ECTS/Workload Table			
Activities	**Count**	**Duration (Hour)**	**Total Workload**
Lesson hours (Including the exam week: 16 x total lesson hours)	16	3	48
Lab			
Application			
Course Specific Internship			
Field Study			
Out of Class Study Time			
Presentation/Seminar Preparation			
Projects			
Reports			
Homework			
Quizzes/Studio Critic			
Preparation Time for Midterm Exams/Midterm Jury	1	40	40
Preparation Time for the General Exam/General Jury	1	62	62
Total Workload	**(ECTS 150/30 = 5)**		**150**