University of Tehran
School of Electrical and Computer Engineering

Course:	**8101420 – Compiler Design and Implementation**
Course type:	<table><tr><td></td><td colspan="6">EE*</td><td colspan="3">CE*</td><td rowspan="2">Credit: 3</td></tr><tr><td></td><td>Com</td><td>E</td><td>P</td><td>B</td><td>Con</td><td>D</td><td>SW</td><td>HW</td><td>IT</td></tr><tr><td>Required</td><td>☐</td><td>☐</td><td>☐</td><td>☐</td><td>☐</td><td>☐</td><td>☒</td><td>☒</td><td>☒</td></tr><tr><td>Elective</td><td>☐</td><td>☐</td><td>☐</td><td>☐</td><td>☐</td><td>☐</td><td>☐</td><td>☐</td><td>☐</td></tr></table>
Level:	Undergraduate ■ Graduate ☐
Co-requisite(s):	None.
Prerequisite(s):	Data Structures (8101437)
Prerequisite by topic:	
Textbook(s):	[1] Alfred V. Aho, Ravi Sethi, and Jeffrey D. Ullman **,** *Compilers: Principles, Techniques, and Tools***,** Second Edition Boston: Addison-Wesley, 2007.
Coordinator:	Ghassemi, Professor, School of ECE
Goals:	In this course, students get familiar with the compiler function in converting a program given in the source language into another program in the target language. Furthermore, they will learn how the basic language constructs are compiled and hence, understand the relation between the features of a source language and the compiler design and implementation. This will help them in future (higher education or work) to develop a compiler for specific operating systems or domain-specific languages (for domains like embedded and real-time). Besides the application, this course is based on the theory of languages that students are familiar with and they will experience one of the applications of this theory in the complier construction.
Outcome:	Upon successful completion of the course, students will be able 1. to implement the components of a compiler by using different implementation techniques, 2. to design a domain-specific language and its compiler, 3. to generate a compiler by automated tools.
Topics:	1) Introduction to different language translators and their differences 2) An abstract introduction to compiler components and their responsibilities 3) Scanner and its implementation: signature and language definition, transition diagram and regular expressions, ANTLR tool, error handling of scanner and its optimization

	4) Parser and its implementation: grammar definition, grammar classification, derivation and parsing concept, non-predictive recursive descent parser, predictive parsers: LL(K), SRL(K), CRL(K), LALR(K), error handling of LL and LR parsers: panic mode, local and global approaches, power of parsers 5) Code analysis techniques exploited during parsing: syntax directed definitions (SDD) and syntax directed schema (SDS), synthesized and inherited properties of languages (regarding the grammar type), SDD application in code conversion / analysis, SDD implementation in recursive descent parser (ANTLR tool), LL and LR, semantic stack 6) Semantics analyzer: the concepts and scope, type checking and approaches, type system and implementation, generation of type expression for arrays, records, functions and objects, subtyping and inheritance, scope management and symbol table 7) Intermediate code generation: expressions, assignment, control flow instruction 8) Management of run-time environment: stack and heap, code generation for procedure definition and calls, arrangement of objects in memory, dynamic lookup in object-oriented languages 9) Intermediate and machine code optimization: control flow analysis, loop and block optimization, peephole optimization
Computer usage:	ANTLR, C++/Java, SPIM
Assignments:	8 Homework
Projects:	Students will implement a compiler for a language with a minimal set of features, designed with the aim of teaching. The result compiler will generate a MIPS code of programs in the source language, which can be run on MIPS simulator. This project is delivered during the term in four phases.
Grading:	Assignments: 10 % Projects: 20 % Quizzes: 10 % Midterm exams: 20 % Final exam: 40 % *It is necessary to achieve at least 50% of midterm and final exams to pass the course.
Further readings:	[1] D. Grune, H. Bal, C. Jacobs, K. Langendoen, *Modern Compiler Design*, John Wiley & Sons, Ltd., 2000 [2] Yunlin Su, and Song Y. Yan, *Principles of Compilers: A New Approach to Compiler*, Springer, 2011.
Prepared by:	Fatemeh Ghassemi Esfahani
Date:	September 4, 2017

*EE: Electrical Engineering		CE: Computer Engineering	
Com	Communications	SW	Software
E	Electronics	HW	Hardware

P	Power	IT	Information Technology
B	Bioelectronics		
Con	Control		
D	Digital System		