

Course Title	Compilers Design				
Course Code	CSE420				
Course Type	Compulsory				
Level	Bachelor (1st Cycle)				
Year / Semester	4 th Year / 7 th Semester				
Teacher's Name	TBA				
ECTS	6	Lectures / week	2 hours/ 14 weeks	Laboratories / week	1 hour/ 14 weeks
Course Purpose and Objectives	To provide knowledge on the structure of Compilers and language translation, terminology in programming analysis, lexical analysis, regular expressions and finite-state techniques. language translation, context free languages, LL and LR parsing methods, syntax directed translation, error recovery, code generation, and portability				
Learning Outcomes	<p>Upon successful completion of the course, students will be able to:</p> <ul style="list-style-type: none"> • Apply basic theoretical principles of compiler design. • Explain and compare basic parsing techniques. • Design basic theoretical machines for compilers. • Describe and compare Run-time Environments: source language issues, storage organization and allocation. • Describe and compare principle sources of optimization. 				
Prerequisites	CSE200, CSE210	Co-requisites	None		
Course Content	<p>Introduction to compilers:</p> <p>Compilers and translators; the structure of a compiler; lexical analysis; syntax analysis; intermediate code generation; optimization code generation; error handling; compiler-writing tools.</p> <p>Programming languages: Definition of programming languages; the lexical and syntactic structure of a language data elements; data structures; operators assignment; statements; program units; data environments; parameter transmission storage management.</p> <p>Finite automata and lexical analysis: The role of lexical analyzer; a simple approach to design of lexical analyzers; regular expressions; finite automata; implementation of lexical analyzer.</p> <p>The syntactic specification of programming languages. Context free grammars; derivatives and parse trees; capabilities of context free grammars.</p> <p>Basic parsing techniques. Parsers; shift-reduce parsing; operator parsing; top down parsing; predictive parsers.</p> <p>Syntax directed translation. Syntax directed translation schemes; intermediate code; postfix notation; parse trees and syntax trees; translation of assignment</p>				

	<p>statements; Boolean expressions; postfix translations; translation with a top-down parser.</p> <p>More about translation. Array references in arithmetic expressions; procedure calls; declarations.</p> <p>Symbol tables. The contents of a symbol table; data structures of symbol tables; representation scope information.</p> <p>Error detection recovery. Code optimization and code generation. The principal source of optimization; object programs; problems in code generation; a simple code generator.</p> <p>Recent developments and contemporary issues pertaining to the subject-matter of the course.</p>										
Teaching Methodology	Face- to- face										
Bibliography	<p>Alfred V. Aho, Monica S. Lam, Jeffrey D. Ullman and Ravi Sethi, COMPILERS: PRINCIPLES, TECHNIQUES, AND TOOLS, Pearson.</p> <p>Aho/Ullman, PRINCIPLES OF COMPILER DESIGN, Addison-Wesley</p> <p>Muchnick, ADVANCED COMPILER DESIGN & IMPLEMENTATION, Morgan Kaufman</p> <p>Weing, F.W., TRANSLATION OF COMPUTER LANGUAGES, Prentice Hall</p>										
Assessment	<table border="1"> <tr> <td>Mid – Term Examination</td> <td>25%</td> </tr> <tr> <td>Final Examination</td> <td>45%</td> </tr> <tr> <td>Assignments/Lab</td> <td>20%</td> </tr> <tr> <td>Class Participation and Attendance</td> <td>10%</td> </tr> <tr> <td></td> <td>100%</td> </tr> </table>	Mid – Term Examination	25%	Final Examination	45%	Assignments/Lab	20%	Class Participation and Attendance	10%		100%
Mid – Term Examination	25%										
Final Examination	45%										
Assignments/Lab	20%										
Class Participation and Attendance	10%										
	100%										
Language	English										