

<b>SYSTEM SOFTWARE AND COMPILERS</b> (Effective from the academic year 2018 -2019) <b>SEMESTER – VI</b>			
<b>Course Code</b>	<b>18CS61</b>	<b>CIE Marks</b>	40
<b>Number of Contact Hours/Week</b>	3:2:0	<b>SEE Marks</b>	60
<b>Total Number of Contact Hours</b>	50	<b>Exam Hours</b>	03
<b>CREDITS –4</b>			
<b>Course Learning Objectives:</b> This course (18CS61) will enable students to:			
<ul style="list-style-type: none"> <li>• Define System Software.</li> <li>• Familiarize with source file, object file and executable file structures and libraries</li> <li>• Describe the front-end and back-end phases of compiler and their importance to students</li> </ul>			
<b>Module 1</b>			<b>Contact Hours</b>
Introduction to System Software, Machine Architecture of SIC and SIC/XE. <b>Assemblers:</b> Basic assembler functions, machine dependent assembler features, machine independent assembler features, assembler design options. Basic Loader Functions <b>Text book 1: Chapter 1: 1.1,1.2,1.3.1,1.3.2, Chapter2 : 2.1 to 2.4, Chapter 3 ,3.1</b> <b>RBT: L1, L2, L3</b>			10
<b>Module 2</b>			
<b>Introduction:</b> Language Processors, The structure of a compiler, The evaluation of programming languages, The science of building compiler, Applications of compiler technology. <b>Lexical Analysis:</b> The role of lexical analyzer, Input buffering, Specifications of token, recognition of tokens. <b>Text book 2:Chapter 1 1.1-1.5 Chapter 3: 3.1 – 3.4</b> <b>RBT: L1, L2, L3</b>			10
<b>Module 3</b>			
Syntax Analysis: Introduction, Context Free Grammars, Writing a grammar, Top Down Parsers, Bottom-Up Parsers <b>Text book 2: Chapter 4 4.1, 4.2 4.3 4.4 4.5</b> <b>RBT: L1, L2, L3</b>			10
<b>Module 4</b>			
Lex and Yacc –The Simplest Lex Program, Grammars, Parser-Lexer Communication, A YACC Parser, The Rules Section, Running LEX and YACC, LEX and Hand- Written Lexers, Using LEX - Regular Expression, Examples of Regular Expressions, A Word Counting Program, Using YACC – Grammars, Recursive Rules, Shift/Reduce Parsing, What YACC Cannot Parse, A YACC Parser - The Definition Section, The Rules Section, The LEXER, Compiling and Running a Simple Parser, Arithmetic Expressions and Ambiguity. <b>Text book 3: Chapter 1,2 and 3.</b> <b>RBT: L1, L2, L3</b>			10
<b>Module 5</b>			
Syntax Directed Translation, Intermediate code generation, Code generation <b>Text book 2: Chapter 5.1, 5.2, 5.3, 6.1, 6.2, 8.1, 8.2</b> <b>RBT: L1, L2, L3</b>			10
<b>Course Outcomes:</b> The student will be able to :			
<ul style="list-style-type: none"> <li>• Explain system software</li> <li>• Design and develop lexical analyzers, parsers and code generators</li> <li>• Utilize lex and yacc tools for implementing different concepts of system software</li> </ul>			

**Question Paper Pattern:**

- The question paper will have ten questions.
- Each full Question consisting of 20 marks
- There will be 2 full questions (with a maximum of four sub questions) from each module.
- Each full question will have sub questions covering all the topics under a module.
- The students will have to answer 5 full questions, selecting one full question from each module.

**Textbooks:**

1. System Software by Leland. L. Beck, D Manjula, 3<sup>rd</sup> edition, 2012
2. Alfred V Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman , Compilers-Principles, Techniques and Tools, Pearson, 2<sup>nd</sup> edition, 2007
3. Doug Brown, John Levine, Tony Mason, lex & yacc, O'Reilly Media, October 2012.

**Reference Books:**

1. Systems programming – Srimanta Pal , Oxford university press, 2016
2. System programming and Compiler Design, K C Louden, Cengage Learning
3. System software and operating system by D. M. Dhamdhare TMG
4. Compiler Design, K Muneeswaran, Oxford University Press 2013.