

SYSTEM SOFTWARE AND COMPILERS (Effective from the academic year 2018 -2019) SEMESTER – VI			
Course Code	18CS61	CIE Marks	40
Number of Contact Hours/Week	3:2:0	SEE Marks	60
Total Number of Contact Hours	50	Exam Hours	03
CREDITS –4			
Course Learning Objectives: This course (18CS61) will enable students to: <ul style="list-style-type: none"> • Define System Software. • Familiarize with source file, object file and executable file structures and libraries • Describe the front-end and back-end phases of compiler and their importance to students 			
Module 1			Contact Hours
Introduction to System Software, Machine Architecture of SIC and SIC/XE. Assemblers: Basic assembler functions, machine dependent assembler features, machine independent assembler features, assembler design options. Basic Loader Functions 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 RBT: L1, L2, L3			10
Module 2			
Introduction: Language Processors, The structure of a compiler, The evaluation of programming languages, The science of building compiler, Applications of compiler technology. Lexical Analysis: The role of lexical analyzer, Input buffering, Specifications of token, recognition of tokens. Text book 2:Chapter 1 1.1-1.5 Chapter 3: 3.1 – 3.4 RBT: L1, L2, L3			10
Module 3			
Syntax Analysis: Introduction, Context Free Grammars, Writing a grammar, Top Down Parsers, Bottom-Up Parsers Text book 2: Chapter 4 4.1, 4.2 4.3 4.4 4.5 RBT: L1, L2, L3			10
Module 4			
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. Text book 3: Chapter 1,2 and 3. RBT: L1, L2, L3			10
Module 5			
Syntax Directed Translation, Intermediate code generation, Code generation Text book 2: Chapter 5.1, 5.2, 5.3, 6.1, 6.2, 8.1, 8.2 RBT: L1, L2, L3			10
Course Outcomes: The student will be able to : <ul style="list-style-type: none"> • Explain system software • Design and develop lexical analyzers, parsers and code generators • Utilize lex and yacc tools for implementing different concepts of system software 			

Question Paper Pattern:
<ul style="list-style-type: none"> • 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:
<ol style="list-style-type: none"> 1. System Software by Leland. L. Beck, D Manjula, 3rd edition, 2012 2. Alfred V Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman , Compilers-Principles, Techniques and Tools, Pearson, 2nd edition, 2007 3. Doug Brown, John Levine, Tony Mason, lex & yacc, O'Reilly Media, October 2012.
Reference Books:
<ol style="list-style-type: none"> 1. Systems programming – Srimanta Pal , Oxford university press, 2016 2. System programming and Compiler Design, K C Loudon, Cengage Learning 3. System software and operating system by D. M. Dhamdhere TMG 4. Compiler Design, K Muneeswaran, Oxford University Press 2013.