

<b>SYSTEM SOFTWARE AND COMPILER DESIGN</b> <b>[As per Choice Based Credit System (CBCS) scheme]</b> <b>(Effective from the academic year 2017 - 2018)</b> <b>SEMESTER – VI</b>			
Subject Code	17CS63	IA Marks	40
Number of Lecture Hours/Week	4	Exam Marks	60
Total Number of Lecture Hours	50	Exam Hours	03
<b>CREDITS – 04</b>			
<b>Module – 1</b>			<b>Teaching 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. <b>Macroprocessors:</b> Basic macro processor functions, <b>Text book 1: Chapter 1: 1.1,1.2,1.3.1,1.3.2, Chapter 2 : 2.1-2.4, Chapter 4: 4.1.1,4.1.2</b>			<b>10 Hours</b>
<b>Module – 2</b>			
<b>Loaders and Linkers:</b> Basic Loader Functions, Machine Dependent Loader Features, Machine Independent Loader Features, Loader Design Options, Implementation Examples. <b>Text book 1 : Chapter 3 ,3.1 -3.5</b>			<b>10 Hours</b>
<b>Module – 3</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, Programming language basics <b>Lexical Analysis:</b> The role of lexical analyzer, Input buffering, Specifications of token, recognition of tokens, lexical analyzer generator, Finite automata. <b>Text book 2: Chapter 1 1.1-1.6 Chapter 3 3.1 – 3.6</b>			<b>10 Hours</b>
<b>Module – 4</b>			
Syntax Analysis: Introduction, Role Of Parsers, Context Free Grammars, Writing a grammar, Top Down Parsers, Bottom-Up Parsers, Operator-Precedence Parsing <b>Text book 2: Chapter 4 4.1 4.2 4.3 4.4 4.5 4.6 Text book 1 : 5.1.3</b>			<b>10 Hours</b>
<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>10 Hours</b>
<b>Course outcomes:</b> The students should be able to: <ul style="list-style-type: none"> <li>• Illustrate system software such as assemblers, loaders, linkers and macroprocessors</li> <li>• Design and develop lexical analyzers, parsers and code generators</li> <li>• Discuss about lex and yacc tools for implementing different concepts of system software</li> </ul>			
<b>Question paper pattern:</b> The question paper will have TEN questions. There will be TWO questions from each module. Each question will have questions covering all the topics under a module. The students will have to answer FIVE full questions, selecting ONE full question from each module.			
<b>Text Books:</b>			

- |                                                                                                                                                                                                                                                                                           |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ol style="list-style-type: none"><li>1. System Software by Leland. L. Beck, D Manjula, 3<sup>rd</sup> edition, 2012</li><li>2. Compilers-Principles, Techniques and Tools by Alfred V Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman. Pearson, 2<sup>nd</sup> edition, 2007</li></ol> |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

<b>Reference Books:</b>
-------------------------

- |                                                                                                                                                                                                                                                                                                                                                       |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ol style="list-style-type: none"><li>1. Systems programming – Srimanta Pal , Oxford university press, 2016</li><li>2. System programming and Compiler Design, K C Loudon, Cengage Learning</li><li>3. System software and operating system by D. M. Dhamdhare TMG</li><li>4. Compiler Design, K Muneeswaran, Oxford University Press 2013.</li></ol> |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|