



SRINIVAS UNIVERSITY
COLLEGE OF ENGINEERING & TECHNOLOGY
 Department Of Computer Science and Engineering
TEACHING/LESSON PLAN (EVEN Semester 2021-22)

Subject Code		19SCS61	Title	SYSTEM SOFTWARE AND COMPILER DESIGN			Class		6 th Semester	
Prerequisites		Operating System			Faculty Name		Mrs. Farha Anjum			
Credits	4	Hours/week	L-T-P: 4		CIE Marks		SEE Marks		Total Hours	50

Course Objectives:

- Define System Software such as Assemblers, Loaders, Linkers and Macro processors.
- 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.
- Describe about language Processors and Lexical Analysis
- Familiarize about syntax Processors

Course Outcomes of the Course:

On Completion of this Course the Student was able to,

CO id	Course Outcome
CO1	Explain system software such as assemblers, loaders, linkers and macro processors.
CO2	Design and develop lexical analyzers, parsers and code generators
CO3	Utilize lex and yacc tools for implementing different concepts of system software
CO4	Analyze about language Processors and Lexical Analysis
CO5	Explain about syntax processors

CO-PO Mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	2									
CO2	3		3									
CO3		2	2									1
CO4	3	2										
CO5	1		3								3	2

Lesson/Teaching Plan of the Course:

Hour No.	Plan Date	Topic to be covered	CO Mapping	Mode of Delivery	Text/ Reference book
1	14/3/22	MODULE 1: Introduction to system software: software/hardware, classification examples		Chalk & Talk	T1
2	15/3/22	System software and machine architecture: difference between system software/ application software		Chalk & Talk	T1
3	18/3/22	Simplified instructional computer (SIC): Various Features of SIC machine architecture		Chalk & Talk	T1
4	19/3/22	SIC Addressing Modes, Instruction set of SIC		Chalk & Talk	T1
5	21/3/22	SIC/XE Machine architecture, features, instruction set, addressing modes		Chalk & Talk	T1

6	22/3/22	Assemblers: Basic assembler functions, types, data structures, assembler algorithm		Chalk & Talk	T1
7	25/3/22	Machine dependent assembler features, instruction formats, program relocation		Chalk & Talk	T1
8	26/3/22	Machine independent assembler features, literals, symbol defining statements		Chalk & Talk	T1
9	28/3/22	Expressions, program blocks, control sections		Chalk & Talk	T1
10	29/3/22	Assembler Design options: one pass assembler, load and go assembler, multi pass assembler		Chalk & Talk	T1
11	1/4/22	MODULE 2: Loaders and Linkers: Basic loader functions,		Chalk & Talk	T1
12	4/4/22	Types of loaders, design of absolute loader		Chalk & Talk	T1
13	5/4/22	A simple bootstrap loader design		Chalk & Talk	T1
14	8/4/22	Loader features, bootstrap loader of SIC/XE machine		Chalk & Talk	T1
15	9/4/22	Machine dependent loader features: Relocation		Chalk & Talk	T1
16	11/4/22	Program linking, Algorithm and data structures		Chalk & Talk	T1
17	12/4/22	Machine independent loader features: Automatic Library search, Loader options		Chalk & Talk	T1
18	16/4/22	Loader design options: Linkage editor, dynamic linking, bootstrap loaders		Chalk & Talk	T1
19	18/4/22	Implementation examples: MSDOS linker,		Chalk & Talk	T1
20	19/4/22	SunOS linkers, Cray MPP linker		Chalk & Talk	T1
21	22/4/22	MODULE 3: Lex and Yacc : The simplest Lex program		Chalk & Talk	T1
22	23/4/22	Recognizing words with Lex, symbol tables		Chalk & Talk	T1
23	25/4/22	Grammars, Parser-lexer communication,		Chalk & Talk	T1
24	26/4/22	The parts of speech lexer, A yacc parser		Chalk & Talk	T1
25	29/4/22	The rules section, Running LEX and YACC,		Chalk & Talk	T1
26	30/4/22	LEX and handwritten lexers examples		Chalk & Talk	T1
27	2/5/22	Using Lex regular expressions,		Chalk & Talk	T1
28	6/5/22	Examples of regular expressions.		Chalk & Talk	T1
29	7/5/22	Examples of regular expressions.		Chalk & Talk	T1
30	9/5/22	A word counting program, parsing a command line		Chalk & Talk	T1
31	10/5/22	MODULE 4: Introduction, Language processors		Chalk & Talk	T2
32	13/5/22	The structure of a compiler: lexical analysis, syntax analysis		Chalk & Talk	T2
33	14/5/22	Applications of compiler technology: Implementation of HLL, optimization for computer architecture		PPT	T2
34	16/5/22	Design of new computer architecture, program translations, Software productivity tools, Program language basics		PPT	T2
35	17/5/22	Lexical analysis: the role lexical analyzer, Input buffering, specification of tokens		PPT	T2

36	20/5/22	Recognitions of tokens: transition diagrams, Architecture of transition diagram based lexical analyzer		PPT	T2
37	21/5/22	Completion of running example, Syntax analysis, introduction		PPT	T2
38	23/5/22	Context free grammars, writing a grammar		PPT	T2
39	24/5/22	Top-down parsing, bottom-up parsing		PPT	T2
40	27/5/22	Simple LR parsers, introduction, viable prefixes		PPT	T2
41	28/5/22	MODULE 5: syntax directed translation, definitions		PPT	T2
42	30/5/22	Evaluation orders of SDD's, dependency graph, semantic rules with controlled side effects		PPT	T2
43	31/5/22	Intermediate code generation: variants of syntax trees		PPT	T2
44	6/6/22	Directed acyclic graphs, constructing DAGs		PPT	T2
45	7/6/22	Three address codes: addresses and instructions, quadruples, triples, static single assignment		PPT	T2
46	10/6/22	Code generation: issues in design of code generator		PPT	T2
47	11/6/22	Input to code generator, target program		PPT	T2
48	13/6/22	Instruction selection, register allocation, evaluation order		PPT	T2
49	14/6/22	The target language, target machine model		PPT	T2
50	17/6/22	Addresses in target code, program and instruction costs		PPT	T2

TEXT/REFERENCE BOOKS:

T/R	BOOK TITLE/AUTHORS/PUBLICATION
T1	System Software by Leland. L. Beck, D Manjula, 3rd edition, 2012
T2	Compilers-Principles, Techniques and Tools by Alfred V Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman. Pearson, 2nd edition, 2007.
R1	Systems programming – Srimanta Pal, Oxford university press, 2016
R2	Compiler Design, K Muneeswaran, Oxford University Press 2013.

Mrs. Farha Anjum

Faculty Member

Date: 14-03-2022

HOD