Course Code:	Course Title	Credit
CSC601	System Programming and Compiler Construction	3

Prerequisite: Theoretical computer science, Operating system. Computer Organization and Architecture .	
Course Objectives:	
1	To understand the role and functionality of various system programs over application programs.
2	To understand basic concepts, structure and design of assemblers, macro processors, linkers and loaders.
3	To understand the basic principles of compiler design, its various constituent parts, algorithms and data structures required to be used in the compiler.
4	To understand the need to follow the syntax in writing an application program and to learn how the analysis phase of compiler is designed to understand the programmer 's requirements without ambiguity
5	To synthesize the analysis phase outcomes to produce the object code that is efficient in terms of space and execution time
Course Outcomes: On successful completion of course, learner will be able to	
1	Identify the relevance of different system programs.
2	Explain various data structures used for assembler and microprocessor design.
3	Distinguish between different loaders and linkers and their contribution in developing efficient user applications.
4	Understand fundamentals of compiler design and identify the relationships among different phases of the compiler.

Module		Content	Hrs
1		**Introduction to System Software**	**2**
	1.1	Concept of System Software, Goals of system software, system program and system programming, Introduction to various system programs such as Assembler, Macro processor, Loader, Linker, Compiler, Interpreter, Device Drivers, Operating system, Editors, Debuggers.	
2		**Assemblers**	7
	2.1	Elements of Assembly Language programming, Assembly scheme, pass structure of assembler, Assembler Design: Two pass assembler Design and single pass Assembler Design for X86 processor, data structures used.	
3		**Macros and Macro Processor**	6
	3.1	Introduction, Macro definition and call, Features of Macro facility: Simple, parameterized, conditional and nested. Design of Two pass macro processor, data structures used.	
4		**Loaders and Linkers**	**6**
	4.1	Introduction, functions of loaders, Relocation and Linking concept, Different loading schemes: Relocating loader, Direct Linking Loader, Dynamic linking and loading.	
5		**Compilers: Analysis Phase**	**10**
	5.1	Introduction to compilers, Phases of compilers: **Lexical Analysis**- Role of Finite State Automata in Lexical Analysis, Design of Lexical analyzer, data structures used.	

		Syntax Analysis- Role of Context Free Grammar in Syntax analysis, Types of Parsers: Top down parser- LL(1), Bottom up parser- SR Parser, Operator precedence parser, SLR. **Semantic Analysis**, Syntax directed definitions.	
6		**Compilers: Synthesis phase**	**8**
	6.1	**Intermediate Code Generation**: Types of Intermediate codes: Syntax tree, Postfix notation, three address codes: Triples and Quadruples, indirect triple. **Code Optimization**: Need and sources of optimization, Code optimization techniques: Machine Dependent and Machine Independent. **Code Generation:** Issues in the design of code generator, code generation algorithm. Basic block and flow graph.	

Textbooks:	
1	D. M Dhamdhere: *Systems programming and Operating Systems,* Tata McGraw Hill, Revised Second Edition
2	A. V. Aho, R. Shethi, Monica Lam, J.D. Ulman: *Compilers Principles, Techniques and Tools*, Pearson Education, Second Edition.
3	J. J. Donovan: *Systems Programming* Tata McGraw Hill, Edition 1991
References:	
1	John R. Levine, Tony Mason & Doug Brown, *Lex & YACC*, O 'Reilly publication, second Edition
2	D, M .Dhamdhere ,*Compiler construction* 2e, Macmillan publication, second edition .
3	Kenneth C. Louden ,*Compiler construction: principles and practices*, Cengage Learning
4	Leland L. Beck, *System software: An introduction to system programming*, Pearson publication, Third Edition
Useful Links for E-resources:	
1	http://www.nptelvideos.in/2012/11/compiler-design.html
2	https://www.coursera.org/lecture/nand2tetris2/unit-4-1-syntax-analysis-5pC2Z

Assessment:	
Internal Assessment:	
Assessment consists of two class tests of 20 marks each. The first -class test is to be conducted when approx. 40% syllabus is completed and the second-class test when an additional 40% syllabus is completed. Duration of each test shall be one hour.	
End Semester Theory Examination:	
1	Question paper will comprise a total of six questions.
2	All question carries equal marks
3	Questions will be mixed in nature (for example supposed Q.2 has part (a) from module 3 then part (b) will be from any module other than module 3)
4	Only Four questions need to be solved.
5	In question paper weightage of each module will be proportional to number of respective lecture hours as mentioned in the syllabus.

Lab Code	Lab Name	Credit
CSL601	System Programming and Compiler Construction Lab	1

Prerequisite: Theoretical computer science, Operating system. Computer Organization and Architecture	
Lab Outcomes: At the end of the course, the students will be able to	
1	Generate machine code by implementing two pass assemblers.
2	Implement Two pass macro processor.
3	Parse the given input string by constructing Top down/Bottom-up parser.
4	Identify and Validate tokens for given high level language and Implement synthesis phase of compiler.
5	Explore LEX & YACC tools.

Suggested List of Experiments	
Sr. No.	**Title of Experiment**
1	Implementations of two pass Assembler.
2	Implementation of Two pass Macro Processor.
3	Implementation of Lexical Analyzer.
4	Implementation of Parser (Any one).
5	Implementation of Intermediate code generation phase of compiler.
6	Implementation of code generation phase of compiler.
7	Study and implement experiments on LEX, YACC.

Reference Books:	
1	Andrew W. Appel Princeton University. Jens Palsberg *Modern Compiler. Implementation in Java*, Second Edition. Purdue University. CAMBRIDGE University press @2002.
2	Charles N. Fischer, Richard J. LeBlanc *Crafting a compiler with C , pearson Education 2007*

Term Work:	
1	Term work should consist of experiments based on suggested experiment list.
2	Journal must include at least 2 assignments on content of theory and practical of "System Programming and Compiler Construction"
3	The final certification and acceptance of term work ensures that satisfactory performance of laboratory work and minimum passing marks in term work.
4	The distribution of marks for term work shall be as follows: Laboratory work (experiments/case studies): .....(15) Marks. Assignment: ..... (05) Marks. Attendance ..... (05) Marks TOTAL: ..... (25) Marks.
Oral & Practical exam will be based on the above and CSC601 syllabus.	