CSI1008	**Principles of Compiler Design**	**L**	**T**	**P**	**J**	**C**
		3	**0**	**0**	**0**	**3**

Pre-requisite	CSI1003	**Syllabus version**
Anti- requisite		

Course Objectives:

1. To provide foundation for study of high performance compilerdesign.
2. To make students familiar with lexical analysisand semantic analysis.
3. To understand the principles of codeoptimization techniques.

Expected Course Outcome:

1. Demonstrate the functioning of a Compiler and to develop a firm and enlightened grasp of concepts suchas higher level programming, assemblers, automata theory, and formal languages, language specifications.
2. Develop language specifications using contextfree grammars(CFG).
3. Apply the ideas, the techniques, and the knowledge acquired for the purpose of developing softwaresystems.
4. Constructing symbol tables and generating intermediate code.
5. Obtain insights on compiler optimization

Student Learning Outcomes (SLO):	**1,2,5**

1.Having an ability to apply mathematics and science in engineering applications.

2. Having a clear understanding of the subject related concepts and of contemporary issues and apply them to identify, formulate and analyse complex engineering problems.

5. Having an ability to use techniques, skills, resources and modern engineering and IT tools necessary for engineering practice

Module:1	**INTRODUCTION TO COMPILATION AND LEXCIAL ANALYSIS**	**7hours**	**CO:1,3**

Introduction to programming language translators-Structure and phases of a compiler-Design issues- Patterns- lexemes-Tokens-Attributes-Specification of Tokens- Extended Regular expression, Regular expression to Deterministic Finite Automata (Direct method)

Module:2	**SYNTAX ANALYSIS –TOP DOWN**	**5 hours**	**CO:2,3**

Role of parser- Parse Tree - Elimination of ambiguity - Top down parsing - Recursive Descent parsing - Non Recursive Descent parsing - Predictive Parsing - LL(1) grammars.

30

Module:3	**SYNTAX ANALYSIS –BOTTOM UP**	**7 hours**	**CO:2,3**

Shift Reduce Parsers- Operator Precedence Parsing ,LR parsers:-Construction of SLR parser tables and parsing , CLR parsing-LALR parsing

Module:4	**SEMANTICS ANALYSIS**	**6hours**	**CO:4**

Syntax Directed Definition – Evaluation Order - Applications of Syntax Directed Translation - Syntax Directed Translation Schemes - Implementation of L attributed Syntax Directed Definition.			
✓ **Module:5**	**INTERMEDIATE CODE GENERATION**	**7hours**	**CO:4**
Variants of syntax trees ✗ - Three address code- Types – Declarations - Procedures - Assignment Statements - Translation of Expressions - Control Flow - Back Patching- Switch Case Statements.			
✓ **Module:6**	**CODE OPTIMIZATION**	**6hours**	**CO:5**
Loop optimizations- Principal sources of optimization -Introduction to Data Flow Analysis - Basic Blocks - The DAG Representation of Basic Blocks -Loops in Flow Graphs.			
✓ **Module:7**	**CODE GENERATION & OTHER TRANSLATIONS ISSUES**	**5hours**	**CO:4,5**
✗ Issues in the design of a code generator- Target Machine- Next-Use Information - Optimization of basic blocks - Peephole Optimization - Register Allocation and Assignment.			
Module:8	**Contemporary issues:**	**2hours**	**CO:1,3**
Recent Trends in Compiler			
	Total Lecture hours:	**45 hours**	
Text Book(s)			
1. 2. 3.	A. V. Aho, Monica S. Lam, Ravi Sethi and Jeffrey D. Ullman, Compilers: principles, techniques, & tools,2007, Second Edition, Pearson Education K. D. Cooper and L. Torczon, Engineering a compiler, Morgan Kaufmann, 2011, 2nd edition. Steven S.Muchnick "Advanced Compiler design implementation", 2003, Elsevier Science India.		
Reference Books			
1. 2. 3.	Andrew A.Appel , Modern Compiler Implementation in Java, Cambridge University Press; 2nd edition, 2002. Allen Holub, Compiler Design in C, Prentice Hall,1990. TorbengidiusMogensen, "Basics of Compiler Design", Springer, 2011.		
Mode of Evaluation: CAT / Assignment / Quiz / FAT / Project / Seminar			
Mode of evaluation:			
Recommended by Board of Studies	DD-MM-YYYY		
Approved by Academic Council	No. xx	Date	DD-MM-YYYY

Sunil