COMPILER DESIGN

Course Code	23CS3601	Year	III	Semester	II
Course Category	PCC	Branch	CSE	Course Type	Theory
Credits	3	L – T – P	3-0-0	Prerequisites	Formal Languages and Automata Theory
Continuous Evaluation:	30	Semester End Evaluation:	70	Total Marks:	100

Course Outcomes		
Upon successful completion of the course, the student will be able to:		
CO1	Understand the phases of a compiler with emphasis on lexical analysis and the fundamental concepts of remaining phases.	L2
CO2	Apply top-down and bottom-up parsing techniques to analyze context-free grammars and construct parsing tables	L3
CO3	Apply syntax-directed definitions and translation schemes for intermediate code generation.	L3
CO4	Apply control-flow and data-flow methods to optimize intermediate code.	L3
CO5	Apply run-time environment mechanisms and code generation techniques	L3

Syllabus		
Unit No.	CONTENTS	Mapped CO
I	**Lexical Analysis:** Language Processors, Structure of a Compiler, Lexical Analysis, The Role of the Lexical Analyzer, Input Buffering, Specification of Tokens, Recognition of Tokens, Lexical Analyzer Generator-LEX, Design of a Lexical Analyzer Generator. **Syntax Analysis:** The Role of the Parser, Context-Free Grammars, Derivations, Parse Trees, Ambiguity, Left Recursion, Left Factoring.	CO1
II	**Top Down Parsing:** Pre Processing Steps of Top Down Parsing, Backtracking, Recursive Descent Parsing, LL (1) Grammars, Non-recursive Predictive Parsing. **Bottom Up Parsing:** Introduction, Difference between LR and LL Parsers, Types of LR Parsers, Shift Reduce Parsing, SLR Parsers, Construction of SLR Parsing Tables, **More Powerful LR Parses**, Construction of CLR (1) and LALR Parsing Tables.	CO1,CO2
III	**Syntax Directed Translation:** Syntax-Directed Definitions, Evaluation Orders for SDD's, Syntax-Directed Translation Schemes. **Intermediate Code Generation:** Variants of Syntax Trees, Three Address Code, Types and Declarations, Translation of Expressions, Type Checking, Control Flow.	CO1, CO3
IV	**Code Optimization:** The Principle Sources of Optimization, Basic Blocks, Optimization of Basic Blocks, Flow Graphs, Loop Optimization, Data-Flow Analysis, Peephole Optimization	CO1, CO4

V	**Run Time Environments:** Storage Organization, Run Time Storage Allocation, Activation Records. **Code Generation:** Issues in the Design of a Code Generator, Code Generation Algorithm, Register Allocation and Assignment.	**CO1,CO5**

Learning Resources
Text Books
1. Compilers: Principles, Techniques and Tools, Second Edition, Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffry D. Ullman, Pearson, 2007.
Reference Books
1. Compiler Construction, Principles and Practice, Kenneth C Louden, Cengage Learning, 2006 2. Modern compiler implementation in C, Andrew W Appel, Revised edition, Cambridge University Press. 3. Optimizing Compilers for Modern Architectures, Randy Allen, Ken Kennedy, Morgan Kauffmann, 2001. 4. Levine, J.R., T. Mason and D. Brown, Lex and Yacc, edition, O'Reilly & Associates, 1990
E-Resources & other digital material
1.http://www.nptel.iitm.ac.in/downloads/106108052/ 2.http://www.vssut.ac.in/lecture_notes/lecture1422914957.pdf