PRASAD V POTLURI SIDDHATHA INSTITUTE OF TECHNOLOGY
(AUTONOMUS)
INFORMATION TECHNOLOGY
AUTOMATA THEORY AND COMPILER DESIGN

Course Code:	23IT3503	**Year:**	III	**Semester:**	I
Course Category:	Professional Core Course	**Branch:**	IT	**Course Type:**	Theory
Credits:	3	**L-T-P:**	3-0-0	**Prerequisites:**	Discrete Mathematics
Continuous Internal Evaluation:	30	**Semester End Evaluation:**	70	**Total Marks:**	100

COURSE OUTCOMES		
Upon successful completion of the course, Student will be able to		
CO1	Recognize the basics of a compiler, formal languages, tokens, and regular expressions.	**L2**
CO2	Construct regular expressions, finite automata, and context-free grammars for recognizing language patterns and solving formal language problems.	**L3**
CO3	Apply the knowledge scanning of tokens and perform the syntax analysis by using parsing techniques	**L3**
CO4	Analyze intermediate code generation and apply basic code optimization techniques to enhance the performance of compiled code.	**L4**

Contribution of Course Outcomes towards achievement of Program Outcomes & Strength of correlations (3:High, 2: Medium, 1:Low)

CO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
CO1	3	2	-	-	-	-	-	-	-	-	-	2	2	1
CO2	3	3	2	-	2	-	-	-	-	-	-	2	2	1
CO3	3	2	3	-	3	-	-	-	-	-	-	2	2	2
CO4	3	3	3	2	3	-	-	-	-	-	-	3	3	3

Unit No.	SYLLABUS CONTENTS	Mapped CO
I	**UNIT – I: Regular Expressions, Languages and Finite Automata** Formal Languages and the Chomsky Hierarchy, Regular Expressions and Regular Languages, Algebraic Laws for Regular Expressions, Applications of Regular Expressions, Abstract model of Finite Automaton, Transition Tables and Transition Graphs, Deterministic Finite Automata (DFA), Nondeterministic Finite Automata (NFA), Converting NFA to DFA, Finite Automata with ϵ transitions (NFA-ϵ), Converting NFA-ϵ to NFA/DFA, Minimization of Finite Automata, Equivalence of FA and Regular Expressions.	CO1,CO2
II	**UNIT-II: Context Free Grammars and Push Down Automata:** Context Free Grammars (CFG) and Context Free Languages (CFL), Design of CFGs, Leftmost and Rightmost Derivations, Parse Trees, Applications of CFGs, Ambiguity in Grammars and Languages, Simplification of Context Free Grammars, Push Down Automata (PDA), The Language of a PDA, Equivalence of PDAs and CFGs, Turning Machine: Definition, Model, and Representation of TMs.	CO1,CO2
III	**UNIT-III: Lexical Analysis and Top-Down Parsing** The structure of a compiler, Role of lexical analyzer, Input Buffering, Specification of tokens, Recognition of tokens, The Lexical Analyser Generator –LEX Introduction to Syntax Analysis, Eliminating ambiguity and left recursion from a CFG, Recursive Decent Parsing, LL(1) Grammars, Nonrecursive Predictive Parsing	CO1,CO3
IV	**UNIT-IV: Bottom-Up Parsing and Syntax Directed Translation** Shift-Reduce Parsing, Simple LR parsing, Canonical LR(1) Parsing, LALR Parsing, Parser Generators Syntax Directed Definitions, Evaluation Orders for SDDs, Syntax Directed Translation Schemes	CO1,CO3
V	**UNIT-V: Intermediate Code Generation, Code Generation and Optimization:** Three address code, Types and Declarations, Translation of Expressions, Type Checking, Control Flow, Issues in the design of a Code Generator, The Target Language, A simple Code Generator Basic Blocks and Flow Graphs, Optimization of Basic Blocks, Peephole Optimization.	CO1,CO4

Learning Resources

Text Books

1. Introduction to Automata Theory, Languages and Computation, J.E.Hopcroft, R.Motwani and J.D.Ullman, 3rd Edition, Pearson, 2008.
2. Compilers Principles, Techniques and Tools, 2nd Edition, Alfred V.Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman, Pearson

References Text Book

1. Introduction to Languages and The Theory of Computation, John C. Martin, McGraw Hill.
2. Theory of Computer Science-Automata, Languages and Computation, K.L.P.Mishra and N.Chandrasekaran, 3rd Edition, PHI, 2007
3. Compiler Construction‖, K.V.N. Sunitha, Pearson, 2013

Compiler Design, SandeepSaxena, Rajkumar Singh Rathore, S.Chand publication