Devi Ahilya Vishwavidhyalaya, Indore, India Institute of Engineering & Technology						II Year B.Tech. (Computer Science & Engineering)
Course Code & Name	**Instructions Hours per Semester and Credits**					
4RCPC5 **Theory of Computation**	**Classroom Instruction (CI)**		**Lab Instruction (LI)**	**Term Work (TW) and Self Learning (SL)**	**Total no. of Hours Per semester**	**Total Credits (Total Hours/30)**
	L	**T**	**P**	**TW+SL**	**90**	**3**
	30	**10**	**-**	**50**		

Prerequisites: Basic knowledge of computer parts, algorithms, flowcharts, operators.

COURSE CONTENTS

UNIT-I

Finite Automata and Regular Languages: Motivation for studying theory of computation, Notion of formal languages and grammars, Kleene's Closure, Regular Expressions and Regular languages, closure properties of regular languages, Finite Automata. Finite Automata with output: Mealy and Moore machines, applications.

UNIT-II

Nondeterminism and Minimization: Nondeterministic Finite Automata,epsilon NFA,conversion between FA types, Acceptance condition.Kleene's Theorem, Myhill-Nerode relations, Minimization Algorithm, Non-Regular languages,Pumping Lemma for regular languages.

UNIT-III

Grammars and Context-Free Languages: Grammars and Chomsky Hierarchy, Context-Free Grammars, Context-Free Languages (CFLs), Inherent Ambiguity of CFLs, closure properties of CFLs, Eliminating useless symbols; null-productions; and unit productions, Chomsky Normal Form, Greibach Normal Form, Cock-Younger-Kasami (CYK) Algorithm, Applications to Parsing.

UNIT-IV

Pushdown Automata: Pushdown Automata (PDAs), PDAs vs CFLs. Deterministic PDAs and CFLs, applications, notion of acceptance for PDAs: acceptance by final states, and by empty stack; the equivalence of the two notions, Proof that CFGs generate the same class of languages that PDAs accept, Pumping Lemma for CFLs.

UNIT-V

Turing Machines and Undecidability: Formal definition of Turing Machines (TM), Variants of TMs, Church-Turing thesis, Universal Turing Machines, Linear Bounded Automata (LBAs) and context-sensitive languages, recursive and recursively enumerable languages, decidability and undecidability, the Halting Problem, Complexity classes (P, NP, NP-complete, NP-hard).

.

Learning Outcomes:

After completing the course, the student will be able to:

1) Model, compare and analyse different computational models.

2) Apply rigorously formal mathematical methods to prove properties of languages, grammars and automata.
3) Identify limitations of some computational models and possible methods of proving them.
4) Have an overview of how the theoretical study in this course is applicable to and engineering application like designing the compilers.
5) Have an understanding of the solvable and unsolvable problems and their computational behaviors.

Course Outcome (CO):

CO. No.	CO	PO
CO1	To introduce core concepts in automata theory, formal languages, computability, and complexity.	PO-1, PO-2, PO-3, PO-4
CO2	To develop the ability to analyze, design, and differentiate between various abstract models of computation	PO-1, PO-2, PO-3, PO-4
CO3	To understand the relationship between different formal language classes (regular, context-free, recursive, recursively enumerable) and their corresponding machine recognizers (FA, PDA, TM).	PO-1, PO-2, PO-3, PO-4
CO4	To rigorously use formal mathematical methods, such as the Pumping Lemma, to prove properties of languages, grammars, and automata, including whether a language is non-regular or non-context-free.	PO-1, PO-2, PO-3, PO-4
CO5	To determine the decidability and intractability of computational problems and understand the limitations of computation, including the Halting Problem and the P vs. NP problem.	PO-1, PO-2, PO-3, PO-4

Books Recommended:

[1]. Daniel I.A. Cohen, Introduction to Computer Theory, John Wiley,Second edition 2007
[2]. John C. Martin, Introduction to Languages and the Theory of Computation, 3/e Tata McGraw Hill, 2005
[3]. J.E. Hopcroft, Rajeev Motwani and J.D.Ullman, Introduction to Automata, Languages and Computation,Pearson Education, Asia, 2008,Third Edition
[4] .Michel Sipser Introduction to Theory of Computation , Cengage Learning,Third Edition
[5]. H.R. Lewis and C.H. Papadimitrou, Elements of the Theory of Computation, Prentice Hall Inc., 1999

CO-PO Relationship:

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	3	3	3	-	-	-	-	-	-	-	1	3	2
CO2	3	3	3	3	-	-	-	-	-	-	-	-	3	2
CO3	3	3	3	3	-	-	-	-	-	-	-	1	2	3
CO4	3	3	3	3	-	-	-	-	-	-	-	2	3	2
CO5	3	3	3	3	-	-	-	-	-	-	-	-	3	2