Semester 3

Course Code	Course Name	Course Type	Cd	L	T	P	Marks		
							Internal	Final Exam	Total
MCA-301	Theory of Computation and Complier Design	PCC	4	4	0	0	40	60	100

Course Outcomes

At the end of the course the student will be able to	
CO1	Apply algorithmic principles and computer science theory to the modeling and design of computational systems.
CO2	Demonstrate knowledge of basic mathematical models of computation and relate them to formal languages.
CO3	Understand the limitations of computers and know about unsolvable problems.
CO4	Understand different phases, intermediate representations, algorithms and principles of working of a compiler
CO5	Apply parsing techniques, error handling and various issues in compiler design for optimized performance.

Detailed Syllabus

Section-A

UNIT–I: Regular languages and Expressions: Symbols, Alphabet, Strings, Backus-Naur Form, Languages, Grammar, Classification of Grammars, Regular Set, Regular expressions, Algebra of Regular expressions, Regular grammar, Regular languages, Closure properties of Regular languages, Applications of regular expressions. Deterministic Finite Automata (DFA) Non-Deterministic Finite Automata (NFA), ε-NFA, Equivalence Finite Automata, Equivalence of Regular Expression and Finite Automata, Pumping Lemma for Regular Languages, Applications of finite automata, Mealy and Moore Machines.

(10Hrs)

UNIT–2: Context Free Grammar**:** Production rules and derivation, Types of Productions, Reduction of Grammar, Chomsky Normal Form, Griebach Normal Form, Pushdown Automata (PDA), Equivalence between CFG and PDA, Context Free Language, Closure properties for context free languages, Pumping Lemma for Context free languages, Applications of Context Free Grammar.

(10Hrs)

UNIT–3: Turing Machines: Description, Transition diagram, Roles of Turing machine, Church-Turing Thesis, Modular Construction of complex Turing machines, Extensions of Turing machines, Non-Deterministic Turing Machines. Universal Turing Machine, Turing acceptable and Turing decidable languages, Decidable and Undecidable Problems, The Halting Problem, Reduction to Another Undecidable Problem, Undecidability of Post Correspondence Problem.

(10Hrs)

Section-B

UNIT–4: Compiler Structure & Front end: Compiler Structure, Compilers and Translators, Analysis- Synthesis Model of Compilation, Various Phases of Compiler Lexical Analysis: Interface with input, parser and symbol table, token, lexeme and patterns, difficulties in lexical analysis, Error Reporting, LEX., Capabilities of Lexical Analyzer, Basic Parsing Techniques: Top-Down parsers with backtracking, Recursive Descent Parsers, Predictive Parsers, Non-recursive Predictive Parsers

(10Hrs)

UNIT–5: Errors Detection and Recovery, Code generation & optimization: Bottom–up Parsers, Shift-Reduce Parsing, Operator Precedence Parsers, LR parsers, Error Detection and Recovery: Lexical phase errors, Syntactic phase errors, Semantic errors, Intermediate Code Generation: Different Intermediate forms: three address code, Quadruples & Triples, Sources of optimization, Local optimization, Peephole optimization, Code Generation: Issues in the design of Code Generator, Basic Blocks and Flow Graphs, Transformations on Basic Blocks, Code , Generation Algorithm, Register Allocation and Assignment.

(10Hrs)