Course Objective: 1. To familiarize with the basic concepts of compiler design such as parsing and code optimization.
2.To design and implement different phases of a compiler.
3. To execute IR and target code generation and optimization.

S. NO.	Course Outcomes (CO)
CO1	Ability to understand and describe the phases of a compiler
CO2	Ability to design and implement lexical analyzer
CO3	Ability to design and implement top-down or LL parsers
CO4	Ability to design and implement bottom-up or LR parsers
CO5	Ability to translate various expressions and statements.
CO6	Ability to execute IR and target code generation and optimization

S. NO.	Contents	Contact Hours
UNIT 1	**Introduction:** Compiler flowchart - Phases of a compiler. Data Structure for Symbol Tables, representing scope information. Run time allocation: Stack versus Heap management.	**6**
UNIT 2	**Lexical analysis**: Input buffers and sentinels, Tokens and lexemes, Lexical categories, Implementation of Lexical Analyzer for an input string using regular expressions and NFA, Thompson algorithm, Subset construction, Automatic Lexical Analyzer Generator - Lex.	**8**
UNIT 3	**Syntax analysis**: Formal Grammars and their application to Syntax Analysis, BNF Notation, Derivation and Parse Trees, Top down parsing- elimination of left recursion, left factoring, recursive descent parsers, predictive parsers or LL(k) parsers. Bottom up parsing- LR Parsers, the canonical collection of LR(0) items, constructing SLR Parsing Tables, canonical LR Parsing tables and LALR parsing tables, An Automatic Parser Generator - YACC. Error detection and error recovery schemes.	**10**

UNIT 4	**Syntax Directed Translation**: Syntax Directed Definition (SDD), L-attributed and S-attributed SDD, Parse Trees, Annotated parse tree and Abstract Syntax Tree (AST), Syntax directed Translation Schemes, Postfix notation, Desktop calculator, Semantic analysis, Translation of Assignment Statements, Boolean expressions, Control Statements, Array references , Procedure Calls, Declarations and Case statements.	**8**
UNIT 5	**Code generation and optimization**: Three address code or IR code, Storage structures for IR code: Quadruple, Triple and Indirect Triple. Syntax directed translation for IR code generation. IR code optimization using Directed Acyclic Graph (DAG), Loop optimization, Global data flow analysis. Target code generation and optimization.	**10**
	TOTAL	**42**

REFERENCES		
S.No.	**Name of Books/Authors/Publishers**	**Year of Publication / Reprint**
1	Aho, Lam, Sethi and Ullman, “Compilers: Principles, Techniques and Tools”, Second edition, Pearson Education.	2013
2	D.M.Dhamdhere, “Compiler Construction – Principles & Practice”, Macmillan India.	2000
3	A. Appel, "Applied Intelligence for Medical Image Analysis", Cambridge university press.	2004
4	K.C. Loude, "Compiler construction: Principles and Practice", Course Technology Inc	1997
5	K.D. Cooper and L. Torczon, "Engineering a compiler", 3rd ed., Morgan Kaufmann.	2023

B.Tech. Information Technology				
Course code: Course Title	**Course Structure**			**Pre-Requisite**
	L	**T**	**P**	Operating systems, Algorithm