CO 1	Acquire knowledge of different phases and passes of the compiler and also able to use the compiler tools like LEX, YACC, etc. Students will also be able to design different types of compiler tools to meet the requirements of the realistic constraints of compilers.	**K_3, K_6**
CO 2	Understand the parser and its types i.e. Top-Down and Bottom-up parsers and construction of LL, SLR, CLR, and LALR parsing table	**K_2,K_6**
CO 3	Implement the compiler using syntax-directed translation method and get knowledge about the synthesized and inherited attributes.	**K_4,K_5**
CO 4	Acquire knowledge about run time data structure like symbol table organization and different techniques used in that.	**K_2, K_3**
CO 5	Understand the target machine's run time environment, its instruction set for code generation and techniques used for code optimization.	**K_2, K_4**
DETAILED SYLLABUS		**3-0-0**

Unit	Topic	Proposed Lecture
I	**Introduction to Compiler:** Phases and passes, Bootstrapping, Finite state machines and regular expressions and their applications to lexical analysis, Optimization of DFA-Based Pattern Matchers implementation of lexical analyzers, lexical-analyzer generator, LEX compiler, Formal grammars and their application to syntax analysis, BNF notation, ambiguity, YACC. The syntactic specification of programming languages: Context free grammars, derivation and parse trees, capabilities of CFG.	08
II	**Basic Parsing Techniques:** Parsers, Shift reduce parsing, operator precedence parsing, top down parsing, predictive parsers Automatic Construction of efficient Parsers: LR parsers, the canonical Collection of LR(0) items, constructing SLR parsing tables, constructing Canonical LR parsing tables, Constructing LALR parsing tables, using ambiguous grammars, an automatic parser generator, implementation of LR parsing tables	08
III	**Syntax-directed Translation:** Syntax-directed Translation schemes, Implementation of Syntaxdirected Translators, Intermediate code, postfix notation, Parse trees & syntax trees, three addresscode, quadruple & triples, translation of assignment statements, Boolean expressions, statementsthat alter the flow of control, postfix translation, translation with a top down parser. More abouttranslation: Array references in arithmetic expressions, procedures call, declarations and casestatements.	08
IV	**Symbol Tables:** Data structure for symbols tables, representing scope information. Run-TimeAdministration: Implementation of simple stack allocation scheme, storage allocation in blockstructured language. Error Detection & Recovery: Lexical Phase errors, syntactic phase errorssemantic errors.	08
V	**Code Generation:** Design Issues, the Target Language. Addresses in the Target Code, BasicBlocks and Flow Graphs, Optimization of Basic Blocks, Code Generator. Code optimization:Machine-Independent Optimizations, Loop optimization, DAG representation of basic blocks,value numbers and algebraic laws, Global Data-Flow analysis.	08