SUBJECT NO-CS31003, SUBJECT NAME- COMPILERS

LTP- 3-0-0,CRD- 3

SYLLABUS :-

The aim is to learn how to design and implement a compiler and also to study the underlying theories. The main emphasis is for the imperative language.Introduction: Phases of compilation and overview.Lexical Analysis (scanner): Regular languages, finite automata, regular expressions, from regular expressions to finite automata, scanner generator (lex, flex).Syntax Analysis (Parser): Context-free languages and grammars, push-down automata, LL(1) gram-mars and top-down parsing, operator grammars, LR(O), SLR(1), LR(1), LALR(1) grammars and bottom-up parsing, ambiguity and LR parsing, LALR(1) parser generator (yacc, bison)Semantic Analysis: Attribute grammars, syntax directed definition, evaluation and flow of attribute in a syntax tree.Symbol Table: Its structure, symbol attributes and management.Run-time environment: Procedure activation, parameter passing, value return, memory allocation, and scope.Intermediate Code Generation: Translation of different language features, different types of intermediate forms.Code Improvement (optimization): Analysis: control-flow, data-flow dependence etc.; Code improvement local optimization, global optimization, loop optimization, peep-hole optimization etc. Architecture dependent code improvement: instruction scheduling (for pipeline), loop optimization (for cache memory) etc.Register allocation and target code generationAdvanced topics: Type systems, data abstraction, compilation of Object Oriented features and non-imperative programming languages.