CS-701 – Compiler Design

Unit-I Introduction to compiling & Lexical Analysis

Unit-II Syntax Analysis & Syntax Directed Translation
Syntax analysis: CFGs, Top down parsing, Brute force approach, recursive descent parsing, transformation on the grammars, predictive parsing, bottom up parsing, operator precedence parsing,LR parsers (SLR, LALR, LR), Parser generation, Syntax directed definitions: Construction of Syntax trees, Bottom up evaluation of S-attributed definition, L-attribute definition, Top down translation, Bottom Up evaluation of inherited attributes Recursive Evaluation, Analysis of Syntax directed definition.

Unit-III Type Checking & Run Time Environment
Type checking: type system, specification of simple type checker, equivalence of expression, types, type conversion, overloading of functions and operations, polymorphic functions. Run time Environment: storage organization, Storage allocation strategies, parameter passing, dynamic storage allocation , Symbol table

Unit –IV Code Generation
Intermediate code generation: Declarations, Assignment statements, Boolean expressions, Case statements, Back patching, Procedure calls Code Generation: Issues in the design of code generator, Basic block and flow graphs, Register allocation and assignment, DAG representation of basic blocks, peephole optimization, generating code from DAG.

Unit –V Code Optimization
Introduction to Code optimization: sources of optimization of basic blocks, loops in flow graphs, dead code elimination, loop optimization, Introduction to global data flow analysis, Code Improving transformations , Data flow analysis of structure flow graph Symbolic debugging of optimized code.

References:
2 Raghavan, Compiler Design, TMH Pub.
5. Mak, writing compiler & Interpreters, Willey Pub.

List of Experiments:
develop a lexical analyzer to recognize a few patterns.
write a programme to parse using Brute force technique of Topdown parsing.
develop LL (1) parser (Construct parse table also).
develop an operator precedence parser (Construct parse table also)
develop a recursive descent parser
write a program for generating for various intermediate code forms i) Three address code ii) Polish notation
write a program to simulate Heap storage allocation strategy
generate lexical analyzer using LEX
generate YACC specification for a few syntactic categories.
given any intermediate code form implement code optimization techniques
study of an object oriented compiler.