COMPILER DESIGN
(Professional Elective-II - Online)

Course Code : 15IT11M1

L	T	P	C
3	0	0	3

Pre-requisites:

Formal Languages & Automata Theory

Course Outcomes:

At the end of the Course, the Student will be able to:

CO 1 Understand the phases in the design of compiler

CO 2 Design top-down and bottom-up parsers

CO 3 Identify synthesis and inherited attributes

CO 4 Develop syntax directed translation schemes

CO 5 Develop algorithms to generate code for a target machine

UNIT-I: (8 Lectures)

COMPILER STRUCTURE:

Analysis-synthesis model of compilation, various phases of a compiler, tool based approach to compiler construction.

LEXICAL ANALYSIS:

Interface with input, parser and symbol table, token, lexeme and patterns, difficulties in lexical analysis, error reporting, and implementation. Regular definition, Transition diagrams, LEX.

UNIT-II: (8 Lectures)

SYNTAX ANALYSIS:

Context free grammars, ambiguity, associativity, precedence, top down parsing, recursive descent parsing, transformation on the grammars, predictive parsing, Bottom up parsing, operator precedence grammars, LR parsers (SLR, LALR, LR), YACC.

UNIT-III: (15 Lectures)

SYNTAX DIRECTED DEFINITIONS:

Inherited and synthesized attributes, dependency graph, evaluation order, bottom up and top down evaluation of attributes, L- and S-attributed definitions.

TYPE CHECKING:

Type system, type expressions, structural and name equivalence of types, type conversion, overloaded functions and operators, polymorphic functions, Run time system: storage organization, activation tree, activation record, parameter passing, symbol table, dynamic storage allocation

UNIT-IV: (9 Lectures)

INTERMEDIATE CODE GENERATION:

Intermediate representations, translation of declarations, assignments, Intermediate Code generation for control flow, boolean expressions and procedure Calls, implementation issues, DAG representation of programs, code generation from dags, peep hole optimization, code generator generators, specifications of machine

UNIT-V: (7 Lectures)

Code optimization, source of optimizations, optimization of basic blocks, loops, global dataflow analysis, solution to iterative dataflow equations, Code improving transformations, dealing with aliases, data flow analysis of structured flow graphs

WEB REFERENCES:

1.   http://nptel.ac.in/courses/106104123
2.   http://nptel.ac.in/courses/106104072/