Advanced Compiler Design

Coursetitle&Code	Credits	Creditdistributionofthecourse			criteria	Pre-requisiteof the course (if any)
		Lecture	Tutorial	Practical/Pract		
Advanced Compiler Design	4	3	0	1	Class XII Pass	NA

COURSE OBJECTIVE

- To understand the fundamental concepts of compilers, including the roles of lexical analysis, parsing, and code generation.
- To gain proficiency in implementing both top-down and bottom-up parsing techniques, including the use of LR and LL grammars.
- To learn the process of syntax-directed translation and the generation of intermediate code representations like three-address code.
- To explore type checking, runtime environments, and how to handle storage allocation, parameter passing, and symbol tables in a compiler.
- To develop skills in code optimization techniques, including basic block optimization, global data flow analysis, and machine-dependent code generation.

COURSE OUTCOMES

At the end of the course, students will be able to:

- Explain and implement the phases of a compiler, including lexical analysis, parsing, and syntax-directed translation.
- Demonstrate the ability to design and implement top-down and bottom-up parsers, including working with LR and LL grammars.
- Gain hands-on experience in generating intermediate code and applying syntax-directed translation techniques.
- Implement type checking, handle runtime environments, and manage memory allocation and symbol tables in a compiler.
- Apply code optimization strategies, such as basic block optimization and global data flow analysis, to improve the efficiency of generated code.

SYLLABUS

Unit 1: Introduction to Compilers & Parsing (9 Hours)
Definition of compiler, interpreter, and their differences, Phases of a compiler, Role of lexical analyzer, Regular expressions, Finite automata, From regular expressions to finite automata, LEX

(lexical analyzer generator), Parsing, Role of parser, Context-free grammar, Derivations, Parse trees, Ambiguity, Elimination of left recursion, Left factoring

Unit 2: Bottom-Up Parsing (8 Hours)
Bottom-up parsing, Definitions and handles, Stack implementation of shift-reduce parsing, Conflicts during shift-reduce parsing, LR grammars, LR parsers (simple LR, canonical LR, LookAhead LR), Error recovery in parsing, Parsing ambiguous grammars

Unit 3: Syntax Directed Translation & Intermediate Code Generation (9 Hours)
Syntax-directed definitions, Construction of syntax trees, S-attributed and L-attributed definitions, Translation schemes, Emitting translations, Intermediate code generation, Types of intermediate forms, Abstract syntax tree, Polish notation, Three-address code, Types of three-address statements, Syntax-directed translation into three-address code

Unit 4: Type Checking & Runtime Environments (9 Hours)
Type checking, Definitions, Type expressions, Type systems, Static vs dynamic checking of types, Specification of a simple type checker, Equivalence of type expressions, Type conversions, Runtime environments, Storage organization, Allocation strategies, Access to non-local names, Parameter passing, Symbol tables, Dynamic storage allocation

Unit 5: Code Optimization & Code Generation (10 Hours)
Code optimization, Organization of the optimizer, Basic blocks, Flow graphs, Optimization of basic blocks, Sources of optimization, Directed acyclic graph (DAG) representation of basic blocks, Global data flow analysis, Code generation, Machine-dependent code generation, Object code forms, Target machines, Simple code generator, Register allocation, Peephole optimization

REFERENCE BOOKS

1. Alfred V. Aho, Ravi Sethi, Jeffrey D. Ullman (2007), *Compilers: Principles, Techniques, and Tools*, 2nd edition, Pearson Education, New Delhi, India.
2. Alfred V. Aho, Jeffrey D. Ullman (2001), *Principles of Compiler Design*, Indian student edition, Pearson Education, New Delhi, India.
3. Kenneth C. Louden (1997), *Compiler Construction: Principles and Practice*, 1st edition, PWS Publishing.
4. K.L. P. Mishra, N. Chandrashekaran (2003), *Theory of Computer Science: Automata, Languages, and Computation*, 2nd edition, Prentice Hall of India, New Delhi, India.