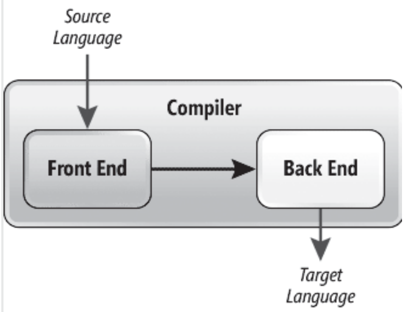
16CS303 COMPILER DESIGN



Hours Per Week :

L	T	P	C
3	-	-	3

Total Hours :

L	T	P	WA/RA	SSH/HSH	CS	SA	S	BS
45	-	-	5	40	2	5	2	2

Course Description and Objectives:

This course introduces the concepts of lexical analyser, parser, code generation and code optimization techniques. The objective of this course is to enable the student to acquire the knowledge of various phases of compiler such as lexical analyser, parser and code optimization.

Course Outcomes:

The student will be able to:

- understand scanner and parser using LEX and YACC tools.
- generate abstract syntax tree.
- derive the three address code from the parse tree.
- understand the code optimization techniques to improve the performance of the compiler.

SKILLS:

- ✓ *Design various parsers using top-down and bottom-up approaches.*
- ✓ *Usage of generators like LEX and YACC.*
- ✓ *Analyse recognizer for programming language.*

ACTIVITIES:

- *Construct the lexical analyser using LEX.*
- *Construct various parsers using YACC.*
- *Design the symbol table.*

UNIT - 1 **L-9**

INTRODUCTION TO COMPILING: Compilers, Analysis of the source program, Phases of a compiler, Cousins of the Compiler, Grouping of Phases, Compiler construction tools, Lexical Analysis, Role of Lexical Analyzer, Input Buffering, Specification of Tokens, Data structures in compilation, LEX, Lexical analyzer generator

UNIT - 2 **L-9**

SYNTAX ANALYSIS: Role of the parser, Writing Grammars, Context-Free Grammars; Top-Down parsing-Recursive Descent Parsing, Predictive Parsing; Bottom-Up parsing-Shift Reduce Parsing, Operator Precedent Parsing; LR Parsers–SLR Parser, Canonical LR Parser, LALR Parser; YACC – automatic parser generator.

UNIT - 3 **L-9**

SEMANTIC ANALYSIS: Intermediate forms of source Programs – abstract syntax tree, polish notation and three address codes; Attributed grammars, Syntax directed translation, Conversion of popular Programming languages, Constructs into Intermediate code forms, Declarations, Assignment, Statements, Boolean Expressions.

UNIT - 4 **L-9**

CODE OPTIMIZATION AND RUN TIME ENVIRONMENTS: Introduction, Principal sources of optimization, Optimization of basic blocks, Introduction to global data flow analysis, Basic blocks, Flow graphs, Data flow equation, Global optimization, Data flow analysis for structured programs.

UNIT - 5 **L-9**

CODE GENERATION: Issues in the design of code generator, The target machine, Next-use Information, A simple Code generator, DAG representation of Basic Blocks, Peephole Optimization.

TEXT BOOK:

1. A.V. Aho, M.S. Lam, R. Sethi and J.D. Ullman, “Compilers: Principles, Techniques and Tools”, 2nd edition, Pearson Education, 2007.

REFERENCE BOOKS:

1. Allen I. Holub, “Compiler Design in C”, 1st edition, Prentice Hall of India, 2003.
2. C. N. Fischer and R. J. LeBlanc, “Crafting a compiler with C”, 1st edition, Benjamin Cummings, 2003.
3. J.P. Bennet, “Introduction to Compiler Techniques”, 2nd edition, Tata McGraw-Hill, 2003.
4. Henk Alblas and Albert Nymeyer, “Practice and Principles of Compiler Building with C”, 3rd edition, Pearson/Prentice Hall India, 2001.
5. Kenneth C. Louden, “Compiler Construction: Principles and Practice”, 1st edition, Thompson Learning, 2003.