



JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA
KAKINADA – 533 003, Andhra Pradesh, India

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

III Year – II Semester				
	L	T	P	C
	3	0	0	3
COMPILER DESIGN				

Course Objectives:

Understand the basic concept of compiler design, and its different phases which will be helpful to construct new tools like LEX, YACC, etc.

Course Outcomes: At the end of the course, student will be able to

- Demonstrate phases in the design of compiler
- Organize Syntax Analysis, Top Down and LL(1) grammars
- Design Bottom Up Parsing and Construction of LR parsers
- Analyze synthesized, inherited attributes and syntax directed translation schemes
- Determine algorithms to generate code for a target machine

UNIT I: Lexical Analysis: Language Processors, Structure of a Compiler, Lexical Analysis, The Role of the Lexical Analyzer, Bootstrapping, Input Buffering, Specification of Tokens, Recognition of Tokens, Lexical Analyzer Generator-LEX, Finite Automata, Regular Expressions and Finite Automata, Design of a Lexical Analyzer Generator.

UNIT II: Syntax Analysis: The Role of the Parser, Context-Free Grammars, Derivations, Parse Trees, Ambiguity, Left Recursion, Left Factoring, **Top Down Parsing:** Pre Processing Steps of Top Down Parsing, Backtracking, Recursive Descent Parsing, LL (1) Grammars, Non-recursive Predictive Parsing, Error Recovery in Predictive Parsing.

UNIT III: Bottom Up Parsing: Introduction, Difference between LR and LL Parsers, Types of LR Parsers, Shift Reduce Parsing, SLR Parsers, Construction of SLR Parsing Tables, More Powerful LR Parsers, Construction of CLR (1) and LALR Parsing Tables, Dangling Else Ambiguity, Error Recovery in LR Parsing, Handling Ambiguity Grammar with LR Parsers.

UNIT III: Syntax Directed Translation: Syntax-Directed Definitions, Evaluation Orders for SDD's, Applications of Syntax Directed Translation, Syntax-Directed Translation Schemes, Implementing L-Attributed SDD's. **Intermediate Code Generation:** Variants of Syntax Trees, Three Address Code, Types and Declarations, Translation of Expressions, Type Checking, Control Flow, Backpatching, Intermediate Code for Procedures.

UNIT IV: Run Time Environments: Storage Organization, Run Time Storage Allocation, Activation Records, Procedure Calls, Displays, **Code Optimization:** The Principle Sources of Optimization, Basic Blocks, Optimization of Basic Blocks, Structure Preserving Transformations, Flow Graphs, Loop Optimization, Data-Flow Analysis, Peephole Optimization

UNIT V: Code Generation: Issues in the Design of a Code Generator, Object Code Forms, Code Generation Algorithm, Register Allocation and Assignment.

Text Books:

1. Compilers: Principles, Techniques and Tools, Second Edition, Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman, Pearson Publishers, 2007.



JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA
KAKINADA – 533 003, Andhra Pradesh, India

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Reference Books:

1. Compiler Construction, Principles and Practice, Kenneth C Louden, Cengage Learning, 2006
2. Modern compiler implementation in C, Andrew W Appel, Revised edition, Cambridge University Press.
3. Optimizing Compilers for Modern Architectures, Randy Allen, Ken Kennedy, Morgan Kaufmann, 2001.
4. Levine, J.R., T. Mason and D. Brown, Lex and Yacc, edition, O'Reilly & Associates, 1990