COMPILER DESIGN

[Revised Credit System]

(Effective from the academic year 2018-19)

SEMESTER - V

Subject Code	**CSE 3151**	**IA Marks**	**50**
Number of Lecture Hours/Week	**03**	**Exam Marks**	**50**
Total Number of Lecture Hours	**36**	**Exam Hours**	**03**

CREDITS - 03

Course objectives: This course will enable students to

- Familiarize students with basic structure of a typical modern compiler.
- Design and implement typical Compilers.
- Use tools for compiler construction.
- Comprehend various optimization techniques.

Module -1	**Teaching Hours**
INTRODUCTION: Language Processors, The Structure of a Compiler- Lexical Analysis, Syntax Analysis, Semantic Analysis, Intermediate Code Generation, Code Optimization, Code Generation, Symbol-Table Management. **Text Book 1:**Chapter 1: 1.1, 1.2.1-1.2.7	**2 Hours**
Module -2	
LEXICAL ANALYSIS: The Role of the Lexical Analyzer, Input Buffering, Recognition of Tokens, The lexical analyzer Generator LEX. **Text Book 1:** Chapter 3: 3.1, 3.2, 3.4, 3.5 **Text Book 2:** Chapter 1,2	**6 Hours**
Module – 3	
SYNTAX ANALYSIS:	**14 Hours**

Introduction, Writing a Grammar- Lexical versus Syntactic Analysis, Introduction to top down and bottom up parsing, Parser generators, Eliminating Ambiguity, Elimination of Left Recursion, Left Factoring, Top-Down Parsing, Bottom-Up Parsing- Reductions, Handle Pruning, Shift-Reduce Parsing, Introduction to LR parsing- Simple LR, Why LR Parsers?, Items and LR(0) Automaton, The LR-Parsing Algorithm, Constructing SLR-Parsing Tables, More Powerful LR parsers- Canonical LR(1) Items, Constructing LR(1) Sets of Items, Canonical LR(1) Parsing Tables. **Text Book 1:** Chapter 4: 4.1, 4.3-4.3.4, 4.4-4.5.3, 4.6-4.6.4, 4.7-4.7.3, 4.9 **Text Book 2:** Chapter 3	
Module-4	
SYNTAX DIRECTED TRANSLATION: Syntax-Directed Definitions, Evaluation Order for SDD's- Dependency Graphs, Ordering the Evaluation of Attributes, Applications of Syntax-Directed Translation -Construction of Syntax Trees. **Text Book 1:** Chapter 5: 5.1, 5.2-5.2.2, 5.3-5.3.1	**3 Hours**
Module-5	
INTERMEDIATE CODE GENERATION AND CODE GENERATION: Variants of Syntax Trees, Three Address Code- Addresses and Instructions, Quadruples, Triples, Types and Declarations- Type Expressions, Type Equivalence, Declarations. Issues in Design of Code Generator, The Target Language, Basic Blocks and Flow Graphs, Optimization of Basic Blocks- The DAG Representation of Basic Blocks, Peephole Optimization. **Text Book 1:** Chapter 6: 6.1– 6.2.3, 6.3-6.3.3, 8.1, 8.2, 8.4, 8.5.1, 8.7	**11 Hours**
Course outcomes:	
After studying this course, students will be able to: 1. Develop a basic compiler using various tools like LEX, YACC etc. 2. Implement various techniques for tokenizing, parsing, optimization, code generation etc .	

Text Books:

1. Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman, *Compilers Principles, Techniques and Tools,* (2e), Pearson Education, 2010.
2. John R. Levine, Tony Manson, Doug Brown, *LEX & YACC*, O Reilly Media, 2012.

Reference Books:

1. Kenneth C. Louden, *Compiler Construction - Principles and Practice,* (1e), Thomson, 2007.
2. Vinu V. Das, *Compiler Design using FLEX and YACC*, Prentice-Hall, 2007.