Subject Code: PCCS-112

Subject Name: Compiler Design

Programme: B.Tech. (CSE)	**L:** 3 **T:** 1 **P:** 0
Semester: 6	**Teaching Hours:** 38
Theory/Practical: Theory	**Credits:** 4
Internal Marks: 40	**Percentage of Numerical/Design/Programming Problems:** 40%
External Marks: 60	**Duration of End Semester Exam (ESE):** 3 hours
Total Marks: 100	**Course Status:** Compulsory

Prerequisites: Knowledge of problem solving using different algorithms and basic programming.

Additional Material Allowed in ESE: [NIL]

On completion of the course, the student will have the ability to:

CO#	Course Outcomes
1.	Apply knowledge of system programming and mathematics to solve problems related to language translation.
2.	Identify, formulate and solve engineering problems in the area of language translation and compiler design.
3.	Formulate machine code by considering the system design components and functionalities involved in compilation.
4.	Inspect runtime structure used to represent constructs of programming language during compilation process.
5.	Use of compiler phases to develop an understanding of their use in building tools used for engineering practice.
6.	Developing an awareness of the functionality and complexity of modern compilers to engage in independent and life-long learning in the broadest context of technological change.

Detailed Contents:

Part A

Introduction to Compiler: Language Processors, The Structure of a Compiler, The Grouping of Phases into Passes, Applications of Compiler Technology, Programming Language Basics. [**3 Hours**]

Lexical Analysis: Role of lexical analyzer, Tokens, Patterns, and Lexemes, Attributes for Tokens, Lexical Errors, Input Buffering, Sentinels, Specification of Tokens, Recognition of Tokens, The Lexical-Analyzer Generator Lex, Finite Automata. [**5 Hours**]

Syntax Analysis: Introduction, Role of the parser, Context-Free Grammars (CFG), Writing a Grammar, Writing a Grammar, Top down parsing –Backtracking, LL(1), Recursive descent parsing, Non-recursive

Predictive Parsing. Bottom-up parsing – Shift reduce parsing, LR parsers, SLR parser. Canonical LR parser, LALR parser, Introduction to The Parser Generator Yacc. [**6 Hours**]

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. [**6 Hours**]

Part-B

Intermediate Code Generation: Variants of Syntax Trees, Three-Address Code, Types and Declarations, Translation of Expressions, Type Checking, Control Flow, Backpatching, Switch-Statements, Intermediate Code for Procedures. [**6 Hours**]

Code Generation: Issues in the Design of a Code Generator, The Target Language, Addresses in the Target Code , Basic Blocks and Flow Graphs, Optimization of Basic Blocks, A Simple Code Generator, Peephole Optimization , Register Allocation and Assignment. [**6 Hours**]

Machine-Independent Optimizations: The Principal Sources of Optimization, Introduction to Data-Flow Analysis, Foundations of Data-Flow Analysis, Constant Propagation, Partial-Redundancy Elimination, Loops in Flow Graphs. [**6 Hours**]

Text Books:

1. Alfred V. Aho, Monica S. Lam, Ravi Seth, Jeffrey D. Ullman, “Compilers, Principles, Techniques, & Tools”, Second Edition, Pearson.

Reference Books:

1. Alfred Aho, Ravi Sethi, Jeffrey D Ullman, “Compilers Principles, Techniques and Tools, Pearson Education Asia, 2003.
2. C. Fischer and R. LeBlanc., “Crafting a Compiler”, Benjamin Cummings, 1991.
3. S. Chattopadhyay, “Compiler Design”, PHI, 2011.
4. C. Holub., “Compiler Design in C”, Prentice-Hall Inc., 1993.
5. Appel., “Modern Compiler Implementation in C: Basic Design”, Cambridge Press,2004.

E-Books and online learning material:

1. https://nptel.ac.in/courses/106/104/106104123/
2. http://index-of.es/Varios-2/Compilers.pdf
3. http://hjemmesider.diku.dk/~torbenm/Basics/basics_lulu2.pdf

Online Courses and Video Lectures:

1. https://nptel.ac.in/courses/106/108/106108113/
2. https://nptel.ac.in/courses/106/104/106104072/
3. https://www.youtube.com/playlist?list=PLrjkTql3jnm-wW5XdvumCa1u9LjczipjA
4. https://www.youtube.com/watch?v=h1LSoF_kUzc
5. https://freevideolectures.com/course/3051/compiler-design