

Semester: VI						
COMPILER DESIGN (Theory & Practice)						
Course Code	:	18CS63		CIE Marks	:	100+50
Credits: L:T:P	:	3:0:1		SEE Marks	:	100+50
Total Hours	:	39L + 35P		SEE Duration	:	3 Hrs + 3 Hrs
Course Learning Objectives: The students will be able to						
1.	Learn basic skill for constructing the compiler which gives the good insight into the algorithms, which have wider applications.					
2.	Gain Knowledge of different forms of language translators that shapes compilers.					
3.	Construct lexical analyser and the parsing methods that are typically used in compilers					
4.	Know about the principle ideas in syntax directed definitions and translations to generate intermediate code for the typical programming languages.					
5.	Understand about the Syntax directed translation, code generation and code optimization.					
Unit – I						7Hrs
Introduction to Compiling and Lexical Analysis Introduction, Language Processors, The structure of Compiler, Evolution of programming Languages. Lexical Analysis- The Role of Lexical Analyzer, Input Buffering, Specifications of Tokens, Recognition of Tokens.						
Unit – II						9Hrs
Syntax Analysis Introduction, Context-free Grammars, Writing a Grammar, Top-down Parsing, Bottom-up Parsing, Introduction to LR Parsing: Simple LR, Most powerful LR parsers (Excluding efficient construction and compaction of parsing tables), Using ambiguous grammars.						
Unit – III						9 Hrs
Lexical –Analyzer and Parser generators Lexical –Analyzer generator Lex, The parser generator YACC, Using YACC with ambiguous grammars, Creating YACC lexical Analyzer with LEX, Error recovery in YACC Syntax-Directed Translation Syntax-Directed Definitions, Evaluation orders for SDD, Application of Syntax Directed Translation.						
Unit – IV						7Hrs
Intermediate Code Generation Variants of Syntax trees, Three address code, Types and Declaration-Type Expressions, equivalence, Declaration, Control flow, Back patching.						
Unit – V						7 Hrs
Code Generation and optimization Issues in the design of Code Generator, The Target Language, Address in the target Code, Basic Blocks and Flow graphs, Optimization of Basic blocks, A Simple Code Generator, Peephole Optimization. Introduction to LLVM compiler and Clang.						

Laboratory Component

Student should be able to design phases of compiler by incorporating following features:

- 1 Writing a scanner, writing predictive parser for a language constructs.
- 2 Experiment with scanner (lex/flex) and parser (yacc/byson) generators
- 3 Writing scanner-parse specification for a simple language constructs.
- 4 Translation of the language constructs to an intermediate form (e.g. three-address code),
- 5 Generation of target code (in assembly language) using compiler construction tools.
- 6 Code improvement and optimization using LLVM compiler.

Course Outcomes: After completing the course, the students will be able to

CO 1:	Understand and explore the fundamental concepts of compiler design and its implementation.
CO 2:	Identify and apply rules for designing various phases of compiler
CO 3:	Analyse the practices adopted in constructing an efficient compiler.
CO 4:	Implement and demonstrate in-depth knowledge of various technologies related to principles, techniques and tools for designing compiler.

Reference Books:

1.	Compilers- Principles, Techniques and Tools, Alfred V Aho, Monica S.Lam, Ravi Sethi, Jeffrey D Ullman; 2 nd Edition, 2013, Pearson Education, ISBN – 10-1-292-02434-8, ISBN – 13- 978-1-292-02434-9.
2.	Compiler Design, Santanu Chattopadhyay, 1 st Edition, 2011, PHI Learning, ISBN-978-81-203-2725-2.
3.	Compiler Construction Principles & Practice, Kenneth C Loudon; Cengage Learning, 1 st Edition, 2009. ISBN – 0534939724.
4.	Crafting a Compiler with C, Charles N. Fischer, Richard J. leBlanc, Jr., 1 st Edition, 2009, Pearson Education, ISBN-13:978-0136067054, ISBN-10: 0136067050.

Continuous Internal Evaluation (CIE); Theory (100 Marks)

CIE is executed by the way of Tests (T), Quizzes (Q), and Experiential Learning (EL). Three tests are conducted for 50 marks each and the sum of the marks scored from three tests is reduced to 50. Minimum of three quizzes are conducted and each quiz is evaluated for 10 marks adding up to 30 marks. All quizzes are conducted online. Faculty may adopt innovative methods for conducting quizzes effectively. The number of quizzes may be more than three also. The marks component for experiential learning is 20.

Total CIE is 50 (T) +30 (Q) +20 (EL) = 100 Marks.

Scheme of Continuous Internal Evaluation (CIE); Practical Test for 50 Marks

The Laboratory session is held every week as per the time table and the performance of the student is evaluated in every session. The average marks (AM) over number of weeks is considered for 30 marks. At the end of the semester a test (T) is conducted for 10 marks. The students are encouraged to implement additional innovative experiments (IE) in the lab and are rewarded for 10 marks. Total marks for the laboratory is 50.

Total CIE is 30(AM) +10 (T) +10 (IE) =50 Marks.

Semester End Evaluation (SEE); Theory (100 Marks)

SEE for 100 marks is executed by means of an examination. The Question paper for the course contains two parts, Part – A and Part – B. Part – A consists of objective type questions for 20 marks covering the complete syllabus. Part – B consists of five main questions, one from each unit for 16 marks adding up to 80 marks. Each main question may have sub