

Compiler Design

Course Code	19CS3601	Year	III	Semester	II
Course Category	Program Core	Branch	CSE	Course Type	Theory
Credits	3	L-T-P	3-0-0	Prerequisites	Formal Languages and Automata Theory
Continuous Internal Evaluation :	30	Semester End Evaluation:	70	Total Marks:	100

Course Outcomes

Upon successful completion of the course, the student will be able to

CO1	Understand the fundamental concepts in Compiler Design	L2
CO2	Apply scanning of tokens to perform the Lexical Analysis and Semantic analysis using attribute grammar	L3
CO3	Apply the various parsing techniques to generate the parse trees.	L3
CO4	Analyze various code optimization techniques for intermediate code forms and Code Generation.	L4

Contribution of Course Outcomes towards achievement of Program Outcomes & Strength of correlations (3:Substantial, 2: Moderate, 1:Slight)

[illegible]

Syllabus		
Unit No.	Contents	Mapped CO
I	Language Processors: Overview of language processing system: – preprocessors – compiler – assembler – Linkers & loaders, difference between compiler and interpreter- structure of a compiler:–phases of a compiler. Lexical Analysis: - Role of Lexical Analysis – Input Buffering – Specification of Tokens – Recognition of Token – The Lexical Analyzer Generator (LEX).	CO1,CO2
II	Syntax Analysis: –Introduction: - Role of a parser – Context Free Grammar – Writing Grammar. Top Down Parsing: – Recursive Descent Parsing-FIRST and FOLLOW- LL(1) Grammar – Non recursive Predictive Parsing- Error Recovery in Predictive Parsing.	CO1,CO3
III	Bottom up Parsing: – Reductions – Handle Pruning - Shift Reduce Parsing – Conflicts During Shift–Reduce Parsing. Introduction to simple LR Parsing: – Why LR Parsers – Model of an LR Parsers — Construction of SLR Tables. More powerful LR parsers: -Canonical LR(1) items ,Construction of CLR (1) parsing table – Construction of LALR Parsing tables.	CO1,CO3
IV	Syntax Directed Translation: Syntax Directed Definitions, Evaluation Orders for SDD's, Applications of Syntax-Directed Translation, Syntax-Directed Translation Schemes for Postfix Translation Schemes –Parser Stack Implementation of Postfix SDT's. Runtime Environment: - Storage organization - Stack allocation – Static allocation - Heap management-Introduction to Garbage Collection. Intermediate code: - Variants of Syntax Trees - Three address code – Quadruples - Triples - Indirect Triples.	CO1,CO2
V	Optimization of Basic Blocks: – DAG representation of basic block. Machine independent code optimization - Common sub expression elimination - Constant folding - Copy propagation -Dead code elimination - Strength reduction - Loop optimization. Machine dependent code optimization: - Peephole optimization – Register allocation - Instruction scheduling - Inter Procedural Optimization - Garbage collection via reference counting.	CO1,CO4

Learning Resources
Text Books
1. Compilers: Principles, Techniques and Tools, Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman, Second Edition, Pearson Education.
References
1. Modern Compiler Implementation in C- Andrew N. Appel, Cambridge University.
2. Principles of compiler design, V. Raghavan, Second edition, 2011, TMH.
3. Compiler Design, Muneeswaran K. First Edition, 2012, Oxford University Press.
e-Resources and other Digital Material

1. <http://www.nptel.iitm.ac.in/downloads/106108052/>
2. http://www.vssut.ac.in/lecture_notes/lecture1422914957.pdf