Compiler Design

Course Code	20CS3601	**Year**	III	**Semester**	II
Course Category	PCC	**Branch**	CSE	**Course Type**	Theory
Credits	3	**L-T-P**	3-0-0	**Prerequisites**	Formal Languages and Automata Theory
Continuous 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 of Compiler Design.	**L2**
CO2	Apply top-down parsing techniques to generate the parse trees.	**L3**
CO3	Apply bottom up parsing techniques to generate parse tree for the given grammar.	**L3**
CO4	Apply various code optimization techniques for intermediate code forms and Code Generation.	**L3**
CO5	Analyze the given grammar and apply suitable parsing techniques.	**L4**

Contribution of Course Outcomes towards achievement of Program Outcomes & Strength of correlations

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	√													
CO2													√	
CO3													√	
CO4	√													
CO5		√							√	√				

Course Content		**Mapped CO**
UNIT-1	**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**
	Syntax Analysis:–Introduction: - Role of a parser – Context Free	

UNIT-2	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,CO2, CO5**
UNIT-3	**Bottom up Parsing:** – Reductions – Handle Pruning - Shift Reduce Parsing – Conflicts During Shift–Reduce Parsing. **Introduction to simple LR Parsing:** – Why LR Parsers – Items and LR(0) Automaton- LR Parsing Algorithm– Construction of SLR parsing Tables.	**CO1,CO3, CO5**
UNIT-4	**More powerful LR parsers**: -Canonical LR(1) items, Constructing LR(1) sets of items, Canonical LR(1) parsing table – Construction of LALR Parsing tables. **Runtime Environment:** - Storage organization – Static versus Dynamic Storage Allocation, Stack allocation of Space –- Heap management **Intermediate code:** - Variants of Syntax Trees - Three address code – Quadruples - Triples - Indirect Triples.	**CO1,CO3, CO4, CO5**
UNIT-5	**Code Generation:-** **Basic Blocks and Flow Graphs**-Basic Blocks, Next use Information, Flow Graphs, Representation of Flow Graph, Loops. **Optimization of Basic Blocks:** – DAG representation of basic block. **Machine independent optimization** – Principle sources of Optimization-Causes of Redundancy, Running example: Quick Sort, Semantic Preserving transformations, Global common sub expressions, copy propagation, dead code elimination, code motion, induction variables, and reduction in strength. **Machine dependent code optimization**:- Peephole optimization – Register allocation and Assignment.	**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.	
Reference Books	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 & other digital material	1. http://www.nptel.iitm.ac.in/downloads/106108052/ 2.http://www.vssut.ac.in/lecture_notes/lecture1422914957.pdf	