



B. Tech.
Semester VI

COMPILER DESIGN
(CE5023)

Effective from June-2023

Syllabus version: 1.00

Subject Code	Subject Title	Teaching Scheme			
		Hours		Credits	
		Theory	Practical	Theory	Practical
CE5023	Compiler Design	3	0	3	0

Subject Code	Subject Title	Theory Examination Marks		Practical Examination Marks	Total Marks
		Internal	External	CIE	
CE5023	Compiler Design	40	60	0	100

Objectives of the course:

- To develop students' ability to understand the principles, algorithms, and data structures involved in the design and construction of compilers such as abstract syntax trees, symbol tables, lexical analysis, top-down and bottom-up parsing, intermediate code generation, code optimization, and stack machine.
- To explain the usage of lexical analyzer and parser generators tools used in compiler construction.

Course outcomes:

Upon completion of the course, the student shall be able to,

- CO1: Understand the fundamental principles in compiler design
- CO2: Identify tokens of a high-level programming language, define regular expressions for tokens, and design and implement a lexical analyser.
- CO3: Design and implement a parser using a typical parsergenerator.
- CO4: Understand various syntax-directed translation techniques and generate intermediate code for any programming language.
- CO5: Understand storage allocation techniques and compiler architecture.
- CO6: Understand and apply various code generation and optimization techniques.

Sr. No.	Topics	Hours
Unit – I		
1	Introduction to Compiler: A simple compiler, Difference between interpreter, Assembler and compiler, Overview and use of linker and loader, Types of compilers, The phases of a compiler, Cousins of the compiler, The grouping of phases, Compiler-construction tools.	5
Unit – II		

2	Lexical Analysis: The role of the lexical analyzer, Input buffering, Specification & recognition of tokens, Deterministic finite automata, Nondeterministic finite automata, From regular expression to an NFA- Λ (Thompson construction method), Converting a regular expression directly to a DFA: Syntax tree method.	8
Unit – III		
3	Syntax Analysis: The role of the parser, Context-free grammars, Writing a grammar, Top-down parsing: Recursive descent parsing, LL(1) parsing, Bottom-up parsing: Operator-precedence parsing, LR parsers – SLRparsing, CLR parsing, LALR parsing, Using ambiguous grammars.	10
Unit – IV		
4	Syntax Directed Translation: Syntax-directed definition, Evaluation order for SDD's, Applications of syntax-directed translation: Construction of syntax trees, Syntax-directed translation: Postfix notation, Synthesized attributes, Simple syntax-directed definition, Tree traversals, Translation schemes	8
	Intermediate Code Generation: Variants of syntax trees, Three-address code, Type checking	
Unit – V		
5	Run-Time Environments: Source language issues, Storage organization, Storage-allocation strategies, Access to non-local names, Parameter passing, Symbol tables, Dynamic storage allocation techniques.	6
Unit – VI		
6	Code Generation: Issues in the design of a code generator, The target language, Basic blocks, And flow graphs, Next-use information, A simple code generator, The DAG representation of basic blocks.	8
	Code Optimization: Peephole optimization, The principal sources of optimization, Loops in flow graphs.	

Text book:

1. A.V. Aho, Monica Lam, Ravi Sethi, J. D. Ullman – “Compilers: Principles, Tools & Techniques” – Addison Wesley, 2nd Edition

Reference books:

1. D. M. Dhamdhere – “Compiler Construction: Principles and Practices” –

Macmillan

(India), 2nd Edition.

2. Allen I. Holub – “Compiler Design in C” – Pearson Education, Prentice Hall of India.
3. C. N. Fischer, R. J. LeBlanc – “Crafting a Compiler with C” - Benjamin Cummings.

Course objectives and Course outcomes mapping:

- To develop students’ ability to understand the principles, algorithms, and data structures involved in the design and construction of compilers such as abstract syntax trees, symbol tables, lexical analysis, top-down and bottom-up parsing, intermediate code generation, code optimization, and stack machine: CO1, CO2, CO3, CO4, CO5, and CO6
- To explain the usage of lexical analyzer and parser generators tools used in compiler construction: CO2, CO3

Course units and Course outcomes mapping:

Unit No.	Unit Name	Course Outcomes					
		CO1	CO2	CO3	CO4	CO5	CO6
1	Introduction to Compiler	✓					
2	Lexical Analysis		✓				
3	Syntax Analysis			✓			
4	Syntax Directed Translation & Intermediate Code Generation				✓		
5	Run-Time Environments					✓	
6	Code Generation & Code Optimization						✓

Programme outcomes:

- PO 1: Engineering knowledge: An ability to apply knowledge of mathematics, science, and engineering.
- PO 2: Problem analysis: An ability to identify, formulates, and solves engineering problems.
- PO 3: Design/development of solutions: An ability to design a system, component, or process to meet desired needs within realistic constraints.
- PO 4: Conduct investigations of complex problems: An ability to use the techniques, skills, and modern engineering tools necessary for solving engineering problems.
- PO 5: Modern tool usage: The broad education and understanding of new engineering techniques necessary to solve engineering problems.

- PO 6: The engineer and society: Achieve professional success with an understanding and appreciation of ethical behavior, social responsibility, and diversity, both as individuals and in team environments.
- PO 7: Environment and sustainability: Articulate a comprehensive world view that integrates diverse approaches to sustainability.
- PO 8: Ethics: Identify and demonstrate knowledge of ethical values in non-classroom activities, such as service learning, internships, and field work.
- PO 9: Individual and team work: An ability to function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
- PO 10: Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give/receive clear instructions.
- PO 11: Project management and finance: An ability to demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
- PO 12: Life-long learning: A recognition of the need for, and an ability to engage in life-long learning.

Programme outcomes and Course outcomes mapping:

Programme Outcomes	Course Outcomes					
	C01	C02	C03	C04	C05	C06
P01	✓	✓	✓	✓	✓	✓
P02		✓	✓		✓	✓
P03		✓	✓		✓	✓
P04						
P05		✓	✓			
P06						
P07						
P08						
P09						
P010						
P011						

P012						
------	--	--	--	--	--	--