

GANPAT UNIVERSITY										
FACULTY OF TECHNOLOGY										
Program		Bachelor of Technology				Branch/ Spec.	Computer Science & Engineering (CBA/BDA/CS/CSE)			
Semester		VII				Version	1.0.0.0			
Effective from Academic Year			2025-26			Effective for the batch Admitted in			June 2022	
Subject code		2CSE701		Subject Name		COMPILER DESIGN				
Teaching scheme					Examination scheme (Marks)					
(Per week)	Lecture (DT)		Practical (Lab.)		Total		CE	SEE	Total	
	L	TU	P	TW						
Credit	3	0	1	0	4	Theory	40	60	100	
Hours	3	0	2	0	5	Practical	30	20	50	
Pre-requisites:										
Discrete Mathematics, Theory of Computation										
Learning Outcome:										
After completion of the course, student will be able to:										
<ul style="list-style-type: none"> ● Learn Compiler phases, passes, and language translation mechanisms. ● Understand the concept of token generation, parsing techniques, symbol table and intermediate code generation. ● Apply the code optimization algorithms. ● Understand various types of errors and error handling techniques. 										
Theory syllabus										
Unit	Content								Hrs	
1	Introduction to Compiling & Lexical Analysis: Translator & its types, Brief introduction of the Compiler, Cousins of the Compiler, Analysis-synthesis model of compilation, Phases of a compiler, Front-end and Back-end of the compiler, Concept of Pass, Lexical analysis: Role of Lexical Analyzer, Input buffering, Specification & Recognition of Tokens, Design of a Lexical Analyzer Generator.								8	
2	Syntax Analysis: CFGs, Parsing Types, Top-down parsing, Brute force approach, recursive descent parsing, transformation on the grammars, predictive parsing, bottom-up parsing, operator precedence parsing, LR parsers (SLR,LALR, LR), Parser generation.								13	

3	Syntax Directed Translation & Run Time Environment: Syntax Directed Definitions, Evaluation Orders for Syntax Directed Definitions, Type Checking, storage organization, Storage allocation strategies, Symbol table, Error Detection & Recovery.	8
4	Intermediate Code Generation & Code Generation: Different Intermediate Forms - Abstract Syntax tree, Polish Notation, 3 address code, Implementation of 3 address code-Quadruple, Triples and Indirect triples, Code Generation: Issues in the design of code generator, Basic block and flow graphs, Register allocation, and assignment, DAG representation of basic blocks, code generation using DAG.	8
5	Code Optimization: Classification of optimization, Principle sources of optimization, Optimization of basic blocks, Peephole Optimization, Loops in flow graphs, Local optimization, Global optimization, Data flow analysis of flow graph.	8

Self-Study Topics

GDB-GNU Project Debugger

Practical content

Practicals will be based on phases of compiler such as Lexical Analyser, left recursion, left factoring, FIRST (), FOLLOW (), parsing, 3-address intermediate, code generator, Code optimization

Mooc Course

Course Name: Principles of Compiler Design

Link: <https://nptel.ac.in/courses/106/108/106108113/>

Text Books

- | | |
|---|---|
| 1 | Compilers: Principles, Techniques and Tools By Aho, Lam, Sethi, and Ullman, Second Edition, Pearson |
|---|---|

Reference Books

- | | |
|---|---|
| 1 | Compilers: Principles, Techniques and Tools By Aho, Sethi, and Ullman, Addison-Wesley |
| 2 | Compiler Design in C By Allen I. Holub, Prentice-Hall/Pearson. |
| 3 | Advanced Compiler Design and Implementation By Muchnick, Morgan and Kaufmann |

Course Outcomes:

COs	Description
CO1	Learn Compiler phases, passes, and language translation mechanisms.
CO2	Understand the concept of token generation, parsing techniques, symbol table and intermediate code generation.
CO3	Apply the code optimization algorithms.
CO4	Understand various types of errors and error handling techniques.

Mapping of CO and PO:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------

C01	3	2	1	0	0	0	0	0	0	0	0	2
C02	2	3	2	2	1	0	0	1	2	2	2	2
C03	2	3	2	2	2	0	0	1	1	1	1	2
C04	2	3	2	2	2	0	0	1	1	1	1	2