GANPAT UNIVERSITY

FACULTY OF ENGINEERING & TECHNOLOGY

Programme	Bachelor of Technology	Branch/Spec.	Computer Engineering/Information Technology
Semester	VII	Version	2.0.0.1
Effective from Academic Year	2021-22	Effective for the batch Admitted in	July-2018
Subject code	2CEIT701	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	-	4	Theory	40	60	100
Hours	3	0	2	-	5	Practical	30	20	50

Pre-requisites:

Theory of Computations

Objectives of the course:

The objectives of this course are to provide you with an understanding of:

- Different stages in the process of compilation
- Identify different methods of lexical analysis
- Design top-down and bottom-up parsers
- Synthesized and inherited attributes
- Syntax directed translation schemes
- Algorithms to generate code for a target machine

Theory syllabus:

Unit	Content	Hrs
1.	**Introduction:** Introduction to translators- Assembler, Compiler, Interpreter, Difference between Compiler and Interpreter, Linker, Loader , one pass compiler, multi pass compiler, cross compiler , The components of Compiler, Stages of Compiler: Front end, Back end, Qualities of Good Compiler.	04
2.	**Lexical Analysis:** The Role of the Lexical Analyzer, Specification of Lexemes, Tokens and Patterns. Recognition of Tokens.	02
3.	**Syntax Analysis:** The Role of the Parser, Types of grammar, CFG, Leftmost derivation , Rightmost derivation, Parse Tree, Restriction on CFG, Ambiguous grammar, Top-Down Parsing, Issues of CFG, Recursive Descent Parser, Construction of Predictive Parsing Table , LL (1) Grammar, String Parsing using M-Table, Bottom-Up Parsing: Handle, Shift-reduce parser, LR parsers: LR (0), SLR (1), LALR (1), CLR (1), String parsing procedure using LR parser, R-R and S-R Conflicts.	17
4.	**Syntax-Directed Translation:** Syntax Directed Definitions, Construction of syntax tree, S-attributed and L-attributed SDDs with example.	04
5.	**Intermediate Code Generation:** Implementation of Three Address Code, Intermediate code for all constructs of programming languages (expressions, if-else, for , while, do….while, switch case, Array, Pointer etc.)	05
6.	**Run-time environment:** Procedure activation, Parameter passing, Value return, Memory allocation, and Scope.	05
7.	**Code generation and optimization:** Introduction to machine code generation and optimization, Simple machine code generation, Examples of machine-independent code optimizations.	06
8.	**Error Detection and Recovery:** Functions of the error handler, Classification of errors: Run time error & Compile time error: Lexical , Syntax & Semantic phase error.	02

Practical content:

- Experiments/Practical/Simulations would be carried out based on syllabus

Text Books:

1. Compilers: Principles, Techniques, and Tools , by A.V. Aho, Monica Lam, Ravi Sethi, and J.D. Ullman.
2. Concept of Compiler Design, By Adesh K. Pandey.

Reference Books:

1. Compiler Design By O G Kakade, 4th Edition.
2. K.C. Louden, Compiler Construction: Principles and Practice, Cengage Learning.

ICT/MOOCs Reference:

1. https://nptel.ac.in/courses/106/108/106108113/
2. https://nptel.ac.in/courses/106/105/106105190/

Course Outcomes:

COs	Description
CO1	Develop the lexical Analyzer for specific grammar
CO2	Design top-down and bottom-up parsers for specific given grammar
CO3	Develop syntax directed translation schemes
CO4	Develop algorithms to generate code for a target machine

Mapping of CO and PO:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	3	1	1	0	0	0	1	0	0	0
CO2	3	3	3	1	1	0	0	0	1	0	0	0
CO3	3	3	3	1	0	0	0	0	1	0	0	0
CO4	3	3	3	1	1	0	0	0	1	0	0	0