

Branch Name:	Bachelor of Technology (Computer Engineering / Information Technology)
Program Code:	ET103/ET105
Course Title:	Compiler Design
Course Code:	2ET1030702T
Pre-requisite Course:	Basic knowledge of Language Theory & Theory of Computation

Course Objectives: This course will help students to understand the key components of a compiler for a high level programming language. Topics include lexical analysis, parsing, type checking, code generation and code optimization.

Teaching and Examination Scheme:

Teaching and Examination Scheme:													
Credit	Teaching Scheme Hours per week				Evaluation Scheme (Marks)								Total (Marks)
					Theory (Marks)				Practical (Marks)				
	L	T	P	Total	SEE		CCA		SEE		CCA		
					Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	
3	3	-	-	3	24	60	16	40	-	-	-	-	100

Course Contents:

Unit No	Topics	Total Hours	Weightage (%)
1	Introduction To Compiler: Definition-Structure-Working of Compiler, Types of Compilers, Cousins of the compiler, Analysis of the Source Program, Components of compiler, Front end and Back end of compiler, Pass structure, Error Detection: Errors encountered in different phases of compiler, Error Recovery.	04	10
2	Lexical Analysis: The Role Lexical Analyzer, Input Buffering, Specification of Tokens, Recognition of Tokens, Finite Automata From a Regular Expression, Optimization of DFA , Lexical analyzer generator: LEX	06	20
3	Syntax Analysis And Semantics Analysis: Understanding Parser, CFG, Types of grammar, Left most and Right most derivation, Parse Tree, Ambiguous grammar, Top-Down Parsing, Issues of Top Down Parsing, Construction of Predictive Parsing Table, LL(1) Grammar, Handle, Bottom-Up Parsing: Shift-Reduce parser, LR parsers, Operator-Precedence Parsing, Parser Generator: YACC. Syntax-Directed Definitions, Annotated Parse Trees.	14	30
4	Intermediate Code Generation And Runtime Environment: Different Intermediate Forms, Storage Organization, Storage-Allocation Strategies, Parameter Passing, Symbol Tables, Dynamic Storage Allocation Techniques	06	20
5	Code Optimization And Generation: Code Optimization Technique, Loop Optimization Technique, Issues in the Design of a Code Generator, Basic Blocks and Flow Graphs, DAG representation of Basic Blocks, Peephole Optimization.	07	20

Reference Books:

1. Compilers: Principles, Techniques and Tools By Aho, Lam, Sethi, and Ullman,
2. Compiler Construction-Principle and Practices –D. M. Dhamdhare, Mcmillian
3. The Theory and Practice of Compiler Writing, J. P. Tremblay and P. G. Sorenson, TMH

List of Open-Source Software/learning website:

- <https://nptel.ac.in/courses/106106237>
- <https://www.geeksforgeeks.org/compiler-design-tutorials/>
- <https://www.udemy.com/course/compiler-design-rmm/>

Course Learning Outcomes (CLO):

Course Outcomes	Description	Bloom's Taxonomy Level
CO1	Acquire knowledge of different phases and passes of the compiler and implement phases using compiler tools.	Understanding (2) Apply (3)
CO2	Design and implement different types of parsers	Create (6)
CO3	Apply syntax-directed translation method to generate intermediate code	Apply (3)
CO4	Analyze data structures used for symbol table and runtime organization	Analyze (4)
CO5	Apply code optimization and code generation techniques	Apply (3)

Mapping of CO with PO and PSO

Course Outcomes	Program Outcomes (POs)												Program Specific Outcomes(PSOs)			
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PSO 3	PSO 4
CO1	H	-	-	-	M	-	-	L	L	-	L	M	M	-	-	-
CO2	M	L	H	M	H	-	L	L	L	H	M	L	-	H	-	-
CO3	H	-	-	-	H	-	-	L	-	L	L	L	-	-	-	-
CO4	M	H	-	H	-	-	-	-	-	-	-	L	-	M	-	-
CO5	M	H	-	H	-	-	-	-	-	-	-	L	-	M	-	-

H: High, M: Medium, L: Low