

Syllabus: CSE 353, Compiler Design

School: SSCSE		Batch: 2023-2027	
Program: B.Tech		Current Academic Year: 2025-2026	
Branch:CSE		Semester: VI	
1	Course Code	CSE353	Course Name: Compiler Design
2	Course Title	Compiler Design	
3	Credits	3	
4	Contact Hours (L-T-P)	3-0-0	
	Course Status	Core	
5	Course Objective	1. To provide students with an overview of the issues that arise in Compiler construction as well as to throw light upon the significant theoretical developments and tools that are deep rooted into computer science. 2. To introduce the major phases of Compiler construction and also its theoretical aspects including regular expressions, context-free grammars, Finite Automata etc.	
6	Course Outcomes	After the successful completion of this course, students will be able to : CO 1:Explain the concepts and different phases of compilation with compile time error handling CO 2:Represent language tokens using regular expressions, context free grammar and finite automata and design lexical analyzer for a language CO 3:Compare top down with bottom up parsers, and develop appropriate parser to produce parse tree representation of the input CO 4: Design syntax directed translation schemes for a given context free grammar. CO 5:Generate intermediate code for statements in high level language, Benefits and limitations of automatic memory management. CO 6:Apply optimization techniques to intermediate code and generate machine code for high level language program	
7	Course Description	To provide students with an overview of the issues that arise in Compiler construction as well as to throw light upon the significant theoretical developments and tools that are deep rooted into computer science.	
8	Outline syllabus		CO Mapping

	Unit 1	Introduction				
	A	Introduction to Compiler, Phases and passes, Bootstrapping, Cross-Compiler				CO1, CO2
	B	Finite state machines and regular expressions and their applications to lexical analysis				CO1, CO2
	C	lexical-analyzer generator, Lexical Phase errors				CO1, CO2
	Unit 2	Parsing Techniques				
	A	The syntactic specification of programming languages: Context free grammars, derivation and parse trees.				CO2, CO3
	B	Basic Parsing Techniques: Parsers, Shift reduce parsing, operator precedence parsing, top down parsing, predictive parsers. Automatic Construction of efficient Parsers: LR parsers, the canonical Collection of LR(0) items, constructing SLR parsing tables				CO2, CO3
	C	Constructing Canonical LR parsing tables, Constructing LALR parsing tables, using ambiguous grammars. Syntactic phase errors and semantic errors.				CO2, CO3
	Unit 3	Syntax Directed Translations And Intermediate Code Generation				
	A	Syntax directed definition, Construction of syntax trees, syntax directed translation scheme				CO4,CO5
	B	Variants of Syntax Trees, Three Address Codes				CO4,CO5
	C	Translation of Expression, Type Checking and control flow.				CO4,CO5
	Unit 4	Symbol table				
	A	Data structure for symbols tables, representing scope information.				CO5
	B	Run-Time Administration: Implementation of simple stack allocation scheme				CO5
	C	Run Time Storage Management				CO5
	Unit 5	Code Generation And Optimization				
	A	Sources of Optimization of basic blocks and flow graphs				CO4,CO6
	B	Basic Blocks, Flow graphs, DAG				CO4,CO6
	C	Global Data Flow Analysis				CO4,CO6
	Mode of examination	Theory				
	Weightage Distribution	CA				ETE
		25%				75%
		Assessment 1 (10)	Assessment 2 (5)	Assignment 1 (5)	Assignment 2 (5)	
	Text book/s*	1. 1.Aho, Sethi, Ulman, compilers Principles, Techniques, and Tools, Pearson Education, 2003				
	Other References	1. Laudon, Principles of Compiler Construction. 2. D. M. Dhamdhare Compiler Construction--Principles and Practice, Macmillan India,				

CO and PO Mapping

S. No.	Course Outcome	Program Outcomes (PO) & Program Specific Outcomes (PSO)
1.	CO 1:Explain the concepts and different phases of compilation with compile time error handling	PO1,PO5,PO6,PO9,PSO1,PSO2
2.	CO 2:Represent language tokens using regular expressions, context free grammar and finite automata and design lexical analyzer for a language	PO1,PO2,PO3, PO4,PO5, PSO1, PSO2
3.	CO 3:Compare top down with bottom up parsers, and develop appropriate parser to produce parse tree representation of the input	PO1,PO2,PO3,PSO1,PSO2
4.	CO 4: Design syntax directed translation schemes for a given context free grammar.	PO1,PO2,PO3, PO4,PO5,PO9, PSO2,PSO3
5.	CO 5:Generate intermediate code for statements in high level language, Benefits and limitations of automatic memory management.	PO1,PO2,PO3, PO4,PO5,PO9,PSO1,PSO2,PSO3
6.	CO6:Apply optimization techniques to intermediate code and generate machine code for high level language program	PO1, PO3,PO4, PO4,PO5,PO9, PSO1,PSO2,PSO3

PO and PSO mapping with level of strength for Course Name Compiler Design (Course Code CSE 353)

Cos	PO1	PO 2	PO3	PO4	PO5	PO 6	PO7	PO8	PO9	PO1 0	PO 11	PS O1	PSO2	PSO3
CO1	3	--	--	--	3	--	--	--	2	--	--	2	1	--
CO2	2	2	3	3	2	--	--	--	--	--	--	3	2	--
CO3	3	3	3	--	--	--	--	--	--	--	--	3	2	--
CO4	1	2	3	3	3	--	--	--	3	--	--	--	3	2
CO5	1	1	2	3	2	--	--	--	3	--	--	1	2	2
CO6	2	--	3	3	2	--	--	--	3	--	--	3	2	3

1-Slight (Low) 2-Moderate (Medium) 3-Substantial (High)
Average of non-zeros entry in following table (should be auto calculated).

Course Code	Course Name	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	P O 7	PO 8	PO 9	PO 10	PO 11	PSO 1	PSO 2	PSO 3
-------------	-------------	------	------	------	------	------	------	-------	------	------	-------	-------	-------	-------	-------

CSE353	Compiler Design	2	1.3	2.3	2	2	0	0-	0	1.8	0	0	2	2	1.2
--------	-----------------	---	-----	-----	---	---	---	----	---	-----	---	---	---	---	-----

Strength of Correlation

1. Addressed to *Slight (Low=1)* **extent**
2. Addressed to *Moderate (Medium=2)* **extent**
3. Addressed to *Substantial (High=3)* **extent**