

A. Course Syllabus with COs, Prerequisites for the Course, Teaching Scheme, List of Books and Reference Books, etc.

Course Syllabus

UNIT 1 LEXICAL ANALYSIS Introduction to different phases of compiler, Alphabets And Tokens In Computer Languages, Representation, Token Recognition And Finite Automata, Implementation, Error Recovery.	8 Hrs.
UNIT 2 PARSERS, SDT Syntax Analysis- Introduction, Role Of Parsers, Context Free Grammars Top Down Parsers, Bottom-Up Parsers, Operator-Precedence Parsing, Semantic analysis- Syntax Directed Translation.	18 Hrs.
UNIT 3 CODE GENERATION AND ASSEMBLER Intermediate code generation and Code optimization, Introduction to System Software, Machine Architecture and m/c level representation of programs, Assemblers- MOT, Data structures in Pass1 and Pass2 assembler, forward and backward referencing, back-patching, target code generation	08 Hrs.
UNIT 4 LOADER AND LINKER Loaders and Linkers: Basic Loader Functions, Machine Dependent Loader Features, Machine Independent Loader Features, Loader Design Options, Implementation Examples.	05 Hrs.
Max. 39 Hrs.	

Course Outcomes (COs)

On completion of the course, student will be able to

- CO1- Explain different phases of compiler.
- CO2- Discuss and compare different parsing algorithms.
- CO3- Illustrate Intermediate code generation.
- CO4- Analyze different types of code optimization techniques.
- CO5- Explain the working of linker and loader.
- CO6- Compare pass1 and pass2 of assembler algorithm.

Pre requisites for the Course: Knowledge of Operating System, Theory of Computation

Teaching Scheme

20CP302T					System Software & Compiler Design					
Teaching Scheme					Examination Scheme					
L	T	P	C	Hrs/Week	Theory			Practical		Total Marks
					MS	ES	IA	LW	LE/Viva	
3	0	0	3	3	25	50	25	-	-	100

List of Books and Reference Books

1. Alfred V Aho, M S. Lam, R Sethi, Jeffrey D. Ullman. Compilers-Principles, Techniques and Tools, Pearson.
2. D. M. Dhamdhare, System software and operating system, TMH
3. Galles David, Modern Compiler Design, Pearson
4. John I Moore, Compiler Design Using Java(R): An Object-Oriented Approach, Softmoore Consulting

B. Academic Calendar

Odd Semester: UG Sem. 1 / 3 / 5 / 7 (FoET & FoLS), PG Sem. 1 / 3 (FoET & FoLS) UG Sem 1 SoL (BA LL.B & LL.B)	
Particulars	Date
Semester Registration & Commencement of classes-FoET & FoLS- 1 st Sem	21 st July, 2025
Semester Registration & Documents Verification Sem 1 SoL (BA LL.B & LL.B)	30 July, 2025
Semester Registration, Department Orientation & Commencement of classes for 3/5/7 Sem – FoET & FoLS	21 st July, 2025
Induction Sem 1 SoL (BA LL.B & LL.B)	1 st & 2 nd August, 2025
Classes begin for Sem 1 SoL (BA LL.B & LL.B)	4 th August, 2025
Evaluation of Rural Internship/CSSI & Evaluation of Industry Orientation, & Evaluation of Industrial Internship	4 th – 8 th August, 2025
Independence Day Celebration	15 th August, 2025
Internal Assessment-I (Quiz, Test, Assignment etc.)** Student mentoring week – I	1 st – 4 th September, 2025
Attendance Review-I & Students Feedback	8 th – 12 th September, 2025
Mid Semester Examination / Project Phase 1 Review	15 th September, 2025 onwards
Rangtaal – Navratri Celebration	26 th September, 2025
Last date of showing evaluated answer books and submission of mid semester exam marks	1 st October, 2025
Declaration of Mid Semester Exam Result	8 th October, 2025
Parent Teacher Meeting	11 th October, 2025
Internal Assessment-II (Quiz, Test, Assignment etc.)** Student mentoring week – II, Attendance Review-II	13 th – 17 th October, 2025
Tesseract – The Science & Technical Fest	10-12 th October, 2025
Diwali break (one week)	20 th – 24 th Oct, 2025
Classes End	14 th November, 2025
Practical Examinations, submission of Term Work and Seminars	17 th November, 2025 onwards
Dissertation presentation for PG for FoLS	
Declaration of Detention list of students	19 th November, 2025
End Semester Examinations – FoET, FoLS & SoL(Regular & Backlog Exams)	24 th November, 2025 onwards
Ph.D. Review	November – December, 2025
Project Phase I Exam for PG program of FoET	8 th – 12 th December, 2025
Last date of showing evaluated answer books and submission of end semester exam marks	19 th December, 2025
Winter Break	22 nd – 26 th December, 2025
Rural Internship for FoLS students	December 2025
Declaration of End Semester Exam Result	2 nd January, 2026

Faculty Timetable

Dr. Meera Thapar Khanna (Div-3)

Computer Science & Engineering

Odd Semester 2025, w.e.f : 21st July 2025[Printed : 16-07-2025]

Day	08:00-08:55	09:00-09:55	10:00-10:55	11:10-12:05	12:10-13:05	13:10-14:05	14:10-15:05	15:10-16:05	16:15-17:10	17:15-18:10
Monday		G6 (20CP302P) E213, CP(5) - P		G7 (24CS203P) E215, CP(3) - P						
Tuesday		G5 (20CP302P) E-303, CP(5) - P		G9G10 (20CP407T) E202, CP(7) - L				G5G6 (20CP302T) E310, CP(5) - L		
Wednesday				G7G8 (20CP407T) E202, CP(7) - L				G5G6 (20CP302T) E310, CP(5) - L		
Thursday		G5G6 (20CP302T) E-301, CP(5) - L		G9 (20CP302P) E216, CP(5) - P						
Friday				G8 (24CS203P) F-103, CP(3) - P			G1G2 (20CP407P) E-305, CP(7) - P			
Saturday										

Office Hours: Tuesday, Friday: 4:00-5:30 PM

C. Lesson Plan (Hour-to-Hour Plan)

Lecture No	Topic
Unit-1 (BW + PPT)	
1	Introduction to language processors, Introduction to different phases of compiler
2	Introduction to different phases of compiler, Symbol table and error handling
3	Roles of lexical analysis, Alphabets And Tokens In Computer Languages, Representation, Token Recognition, Introduction to Finite Automata (FA)- RE to FA
4	Conversion from NFA to DFA, conversion from NFA-null to DFA (subset construction)
5	RE to DFA using syntax tree method
6	Minimization of DFA, Lexical errors, error recovery, Input buffering
Unit-2 (BW + PPT)	
7	Syntax Analysis- Introduction, role of parsers, lexer-parser communication
8	Context Free Grammars, ambiguity-arithmetic, dangling else
9	Elimination of left recursion, left factoring
10	Introduction of Top-down parser, Recursive Descent Parser, Introduction of Predictive parser (LL)
11	Predictive parser –first and follow
12	Predictive parser – Table construction
13	LL(1) Parser, parsing the string, Properties of LL(1) grammar
14	Introduction of Bottom up parser, handle, handle pruning
15	Operator precedence parser table
16	Operator precedence parser algorithm, error recovery
17	LR parser- SLR parsing items generation
18	SLR parsing table
19	SLR parsing algorithm
20	CLR parsing items generation
21	CLR parsing table
22	LALR parsing
23	Semantic Analysis - introduction
24	Syntax direct translation, Comparison between SDD and SDT
25	SDD and SDT schemes examples
Unit-3 (BW + PPT)	
26	Intermediate code generation – types of IC
27	Intermediate code generation – memory representations
28	Intermediate code generation – for various programming constructs
29	Basic block, flow graph
30	Code optimization techniques
31	Introduction to System Software, Machine Architecture and m/c level representation of programs
32	Assembler and instruction code
33	Assembler- pass 1 and data structures
34	Assembler- pass 2, Forward and backward referencing, Back-patching
Unit-4 (BW + PPT)	
35	Loaders and Linkers: Basic Loader Functions
36	Machine Dependent Loader Features, Machine Independent Loader Features
37	Loader Design Options, Implementation Examples
38	Types of linkers and comparison
39	Revision

D. Actual Engagement Sheet

As per attachment

E. Mapping of Course Outcomes with Programme Outcomes (PO)

Course Articulation Matrix

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PS01	PSO2	PSO3
CO1	3	-	-	-	-	-	-	-	-	1	-	3	3	1	3
CO2	3	1	2	-	-	1	-	-	1	1	-	3	3	1	3
CO3	3	1	2	-	1	1	-	-	1	1	1	3	3	1	3
CO4	3	2	2	1	1	1	-	1	1	1	1	3	3	1	3
CO5	3	1	1	-	1	1	-	1	1	1	1	3	3	1	3
CO6	3	-	-	-	1	1	-	-	1	1	1	3	3	1	3

Program Articulation Matrix

PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PS01	PSO2	PSO3
3	0.83	1.17	0.17	0.67	0.83	0	0.33	0.83	1	0.67	3	3	1	3

Correlation levels 1, 2 or 3 as defined below:

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

F. Evaluation Scheme and Rubrics

Co Assessment Tools (Direct Assessment):

Various assessment tools used to evaluate CO's (Rubrics) and the frequency with which the assessment processes are carried out are listed below.

Assessment Method	Assessment Tool	Description	Marks	Mapping with CO	Contribution to COs
Direct (MID-Sem Examination)	Descriptive	Descriptive based syllabus covered	10	CO1,CO2	It fractionally contributes to 25% weightage of Direct Assessment to CO attainment.
	Problem solving/ design	Analytical/design/example based questions on syllabus covered	15		
Total 50 marks will be converted into 25 marks for the mid-sem evaluation.					
Direct (Internal Evaluation)	Quiz-1	For each unit a separate assignment will be prepared (Descriptive and analytical/design based questions)	10	CO1,CO2, CO3,CO4	It fractionally contributes to 25% weightage of Direct Assessment to CO attainment.
	Quiz-2 /Assignment	At the end of semester Question-answer based evaluation on One-to-one basis	15	CO3,CO4, CO5, CO6	
Direct	End-Sem Examination	Topics to be covered: Unit I, II, III, IV	100	CO1,CO2, CO3,CO4, CO5, CO6	It contributes to 50% weightage of Direct Assessment to CO attainment.
Total 100 marks will be converted into 50 marks at the end.					

G. Course Presentations (PPTs) Sample Class Notes, Handouts, Course Material, Web links, Software, E-books, Relevant NPTEL and MOOC, Video Lectures

E-books:

<https://www3.nd.edu/~dthain/compilerbook/compilerbook.pdf>

<https://holub.com/goodies/compiler/compilerDesignInC.pdf>

NPTEL:

Compiler Design By Prof. Santanu Chattopadhyay, IIT Kharagpur

https://onlinecourses.nptel.ac.in/noc23_cs57/preview

Other course:

Compilers by Alex Aiken

<https://online.stanford.edu/courses/soe-yescs1-compilers>

Video Lectures:**Chapter-1 of Compiler Design****1. Introduction to Compiler Design**

https://www.youtube.com/watch?v=bU7JdOUucFg&list=PLlQYHyEFKWT-cXv_Bn4wWqt5SMNqSY72h&index=7

2. Phases of Compiler Design- Part I

https://www.youtube.com/watch?v=lUdyzjkgAcs&list=PLlQYHyEFKWT-cXv_Bn4wWqt5SMNqSY72h&index=8

3. Phases of Compiler Design- Part II

https://www.youtube.com/watch?v=eBe4ZrG1Uo8&list=PLlQYHyEFKWT-cXv_Bn4wWqt5SMNqSY72h&index=9

4. Phases of Compiler Design- Part III

https://www.youtube.com/watch?v=-LiHzgmUm94&list=PLlQYHyEFKWT-cXv_Bn4wWqt5SMNqSY72h&index=10

Chapter-2 Lexical Analysis**1. RE to DFA using syntax tree method**

<https://www.youtube.com/watch?v=kDcHklw5Tqo>

2. Recursive Descent Parser:

<https://www.youtube.com/watch?v=Bc8bOWIV83w>

H. List of National/International Journals related to the Course

ACM Transactions on Architecture and Code Optimization

I. List of well-known conferences related to the Course

CC - International conference on Compiler Construction

PLDI – ACM SIGPLAN Conference on Programming Language Design and Implementation

J. List of Classical Journal Papers/Articles/Review Papers related to Course

1. Knuth, D. E. (July 1965). "On the translation of languages from left to right". Information and Control. 8 (6): 607–639. doi: 10.1016/S0019-9958(65)90426-2
2. Knuth, Donald Ervin. "Semantics of context-free languages." Mathematical systems theory 2 (1968): 127-145.
3. Gary Kildall "A Unified Approach to Global Program Optimization" Proceedings of ACM SIGACT-SIGPLAN 1973 Symposium on Principles of programming Languages.
4. Frances E. Allen, J. Cocke "A program data flow analysis procedure" Commun. ACM, 19, 137–147, 1976

K. List of Renowned Industries/Organizations working in the Course related areas

NITs , IITs, MIT

Also, the course demonstrates the problem-solving ability and comparison of algorithms for various applications. Hence, the industries working on software and application developments use the concept of this course directly or indirectly.

L. List of Renowned Scientists/Academicians working in the Course related areas

1. Uday Reddy Bondhugula, Department of Computer Science and Automation, Indian Institute of Science
2. Alfred Aho – compilers book, the 'a' in AWK
3. Bruce Arden – programming language compilers (GAT, Michigan Algorithm Decoder (MAD)), virtual memory architecture, Michigan Terminal System (MTS)
4. Robert M. Graham – programming language compilers (GAT, Michigan Algorithm Decoder (MAD)), virtual memory architecture, Multics
5. Susan L. Graham – compilers, programming environments
6. John Hopcroft – compilers
7. Admiral Grace Hopper (1906–1992) – developed early compilers: FLOW-Matic, COBOL; worked on UNIVAC; gave speeches on computer history, where she gave out nano-seconds
8. Ravi Sethi – compilers, 2nd Dragon Book
9. Jeffrey D. Ullman – compilers, databases, complexity theory