

Consolidated Academic Administration Plan for the Course
System Programming and Compiler Construction (core) Sem. VI –
Program Computer Engineering 2023-24 –Even Semester
Faculty – Pankaj Vanwari (Cluster Mentor)

The academic resources available in VIT –

VMIS (ERP)	V-Refer and V-Live	VIT Library	VAC & MOOC Courses
Institute & Department Vision and Mission	Former IA question papers and solutions (prepared by faculty)	Former IA question papers solutions - hardcopy	Value Added Courses (VAC) are conducted throughout the semester & in the semester break - Enrol for the VACs
Program Educational Objectives (PEO)	MU end semester examination question papers and solutions (prepared by faculty)	MU end semester exam question paper & solutions - by faculty, hardcopy	
Program Specific Outcome (PSO)	Class notes and Digital Content for the subject (scanned / typed by faculty)	All text books, reference books, e -books mentioned in the syllabus & AAP	Online courses from NPTEL, Coursera etc. are pursued throughout the semester - Register for the course & get certified
Program Outcome (PO)	Comprehensive question bank, EQ, GQ, PPT, Class Test papers	Technical journals and magazines for reference	
Departmental Knowledge Map	Academic Administration Plan & Beyond Syllabus Activity report	VIT library is member of IIT Bombay Library	Watch former lectures captured in LMS at VIT

1.a Course Objectives (Write in detail – as per NBA guidelines)

Cognitive	What do you want students to know?	Students should know the procedures for the designing of software systems and to understand the compilation process.
Affective	What do you want students to think / care about?	Students should take care about carrying out optimization on the immediate code generated with appropriate semantic actions and implementations.
Behavioural	What do you want students to be able to do?	Students should be able to solve problems universally encountered in designing a language translator, regardless of the source or target machine.

Advice to Students:

Attend every class!!! Missing even one class can have a substantial effect on your ability to understand the course. Be prepared to think and concentrate, in the class and outside. I will try to make the class very interactive. Participate in the class discussions. Ask questions when you don't understand something. Keep up with the class readings. Start assignments and homework early. Meet me in office hour to discuss ideas, solutions or to check if, what you understand is correct.

The v-Refer Link:

<http://vidyalankarlive.com/vrefer/index.php/apps/files/?dir=/vRefer/CMPN/SEM%20VI/2023-24/SPCC/PV&fileid=768623>

MS Teams link

[AY23-24 Even SPCC PV | General | Microsoft Teams](#)

Collaboration Policy:

We encourage discussion between students regarding the course material. However, no discussion of any sort is allowed with anyone on the assignment and homework for the class. If you find solution to some problems in a book or on the internet, you may use their idea for the solution; provided you acknowledge the source (name and page in the book or the website, if the idea is found on the internet). Even though you are allowed

to use ideas from another source, you must write the solution in your own words. If you are unsure whether or not certain kinds of collaboration is possible please ask the teacher.

1.b Course Outcome (CO) Statements and Module-Wise Mapping (follow NBA guideline)

CO No.	Statements	Related Module/s
CO1	Identify the relevance of different system programs	1
CO2	Describe the various data structures and passes of assembler design	2
CO3	Identify the need for different features and designing of macros	3
CO4	Distinguish different loaders and linkers and their contribution in developing efficient user applications	4
CO5	Construct different parsers for given context free grammars	5
CO6	Justify the need of synthesis phase to produce object code optimized in terms of high execution speed and less memory usage	6

1.c Mapping of COs with POs (mark S: Strong, M: Moderate, W: Weak, Dash '-': not mapped) (List of POs is available in V-refer)

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	M	M	W	-	S	-	-	-	-	-	-	-
CO 2	M	S	S	M	-	-	-	-	-	-	-	-
CO 3	M	M	S	M	-	-	-	-	-	-	-	-
CO 4	M	M	M	S	-	-	-	-	-	-	-	-
CO 5	M	S	M	S	M	-	-	-	-	-	-	-
CO 6	M	M	S	M	-	-	-	-	-	-	-	-

1.d Mapping of COs with PSOs (mark S: Strong, M: Moderate, W: Weak, Dash '-':not mapped)

	PSO 1	PSO 2	PSO 3
CO 1	S	W	-
CO 2	S	S	-
CO 3	M	S	-
CO 4	M	M	-
CO 5	M	S	W
CO6	M	-	-

1.e Teaching and Examination Scheme (As specified by the University) for the Course

Categories	Humanities and Social Sciences	Basic Science	Engineering Science	Professional Core	General Education	Professional Elective	Project/ Internship	Open Elective
------------	--------------------------------	---------------	---------------------	-------------------	-------------------	-----------------------	---------------------	---------------

Tick suitable category				√				
------------------------	--	--	--	---	--	--	--	--

Subject Code	Subject Name	Teaching Scheme			Credits Assigned			
		Theory	Practical	Tutorial	Theory	TW/Practical	Tutorial	Total
CSC601	System Programming and Compiler Construction	3	-		3	-		3
CSL601	System Software Lab	-	2		-	1		1

Subject Code	Subject Name	Examination Scheme					
		Theory			Practical/Oral		Total
		ISA	MSE	ESE	ISA	ESE	
CSC601	System Programming and Compiler Construction	20	30	50	-	-	100
CSL601	System Software Lab	-	-	-	25	25	50

1.f Faculty-Wise Distribution of all Lecture-Practical-Tutorial Hours for the Course

Divisions	Lecture (Hrs.)	Practical (Hrs.)				Tutorial (Hrs.)			
		Batch 1	Batch 2	Batch 3	Batch 4	Batch 1	Batch 2	Batch 3	Batch 4
A	PV (3)	PV (2)	PV (2)	PV (2)	PV (2)	-	-	-	-
B	PV (3)	PV (2)	PV (2)	RSG (2)	RSG (2)	-	-	-	-

1.g Office Hours (Faculty will be available in office in this duration for solving students' query)

Division	Day	Time (at least 1 Hr. / Division)	Venue (Office Room No.)
A	Tuesday	3.45 to 4.45pm	M-209
B	Wednesday	3.45 to 4.45pm	M-209

2.a Syllabus : Module Wise Teaching Hours and % Weightage in University Question Paper

Module No.	Module Title and Brief Details	Teaching Hrs. for each module	% Weightage in University Question Papers
1	Introduction to System Software Concept of System Software, Goals of system softwares, system program and system programming, Introduction to various system programs such as Assembler, Macro processor, Loader, Linker, Compiler, Interpreter, Device Drivers, Operating system, Editors, Debuggers.	2	5

2	Assemblers Elements of Assembly Language programming, Assembly scheme, pass structure of assembler, Assembler Design: Two pass assembler Design and single pass Assembler Design for Hypothetical / X86 family processor, data structures used.	7	18
3	Macros and Macro Processor Introduction, Macro definition and call, Features of Macro facility: Simple, parameterized, conditional and nested. Design of single pass macro processor, data structures used.	6	15
4	Loaders and Linkers: Introduction, functions of loaders, Relocation and Linking concept, Different loading schemes: Relocating loader, Direct Linking Loader, Dynamic linking and loading.	6	15
5	Compilers: Analysis Phase: Introduction to compilers, Phases of compilers: Lexical Analysis- Role of Finite State Automata in Lexical Analysis, Design of Lexical analyser, data structures used . Syntax Analysis- Role of Context Free Grammar in Syntax analysis, Types of Parsers: Top down parser- LL(1), Bottom up parser- Operator precedence parser, SLR Semantic Analysis, Syntax directed definitions	10	26
6	Compilers: Synthesis phase: Intermediate Code Generation: Types of Intermediate codes: Syntax tree, Postfix notation, Three address codes: Triples and Quadruples. Code Optimization: Need and sources of optimization, Code optimization techniques: Machine Dependent and Machine Independent. Code Generation: Issues in the design of code generator, code generation algorithm. Basic block and flow graph	8	21
Total		39	100

2.b Prerequisite Courses

No.	Semester	Name of the Course	Topic/s
1	II	Structured Programming Approach	C programming concept
2	V	Theoretical Computer Science	Finite Automata, Grammar Concept

2.c Relevance to Future Courses

No.	Semester	Name of the Course
1	VII	Natural Language Processing

2.d Identify real life scenarios/examples which uses the knowledge of the subject (Discussion on how to prepare examples and case studies e.g. [“Boeing Plane”: C Programming Language – Intro to Computer Science – Harvard’s CS50 \(2018\) – Bing video](#))

Real Life Scenario	Concept Used
Cross-Assemblers / Compilers	Assembler and compiler design

System Software for java enabled mobile handsets such as an editor, debugger, compiler, etc.. for development on mobile devices	General purpose loading and linking
Tuning gcc optimizer for a dedicated set of application programs.	Optimization techniques

3 Past Results – Division-Wise

Details	Target – May 2024	May 2023	May 2022	May 2021
Course Passing % – Average of 2 Divisions	100	100	100	100
Marks Obtained by Course Topper (mark/100)	96	92	78/80	79/80

	Division A		Division B	
Year	Initials of Teacher	% Result	Initials of Teacher	% Result
May 2023	RSG	100	PJP	100
May 2022	RSG	98.71	RSG	97.47
May 2021	RSG	100	RSG	100

4 All the Learning Resources – Books and E-Resources

4.a List of Text Books (T – Symbol for Text Books) to be Referred by Students

Sr. No	Text Book Titles	Author/s	Publisher	Edition	Module Nos.
1	Systems programming	D. M Dhamdhare	Tata McGraw Hill	Second Edition	1,2,3,4
2	Compilers Principles, Techniques and Tools	A. V. Aho, R. Shethi, Monica Lam , J.D. Ulman	Pearson Education	Second Edition	5,6
3	Systems Programming	J. J. Donovan	Tata McGraw Hill Publishing Company	Second Edition	1,2,3,4

4.b List of Reference Books (R – Symbol for Reference Books) to be Referred by Students

Sr. No	Reference Book Titles	Author/s	Publisher	Edition	Module Nos.
1	Lex &yacc	John R. Levine, Tony Mason & Doug Brown	O'Reilly	II	5,6
2	Compiler construction	D,M.Dhamdhare	MACMILLAM	II	5,6
3	Compiler construction : principles and practices	Kenneth C.Louden	CENGAGE Learning	II	5,6
4	System software : An introduction to system programming	Leland L. Beck	Pearson	II	1,2,3,4

4.c List of E - Books (E – Symbol for E-Books) to be Referred by Students

Sr. No	E- Book Titles	Author/s	Publisher	Edition	Module Nos.
1	Compiler Construction	William Waite and Gerhard Goos	Springer	-	5,6
2	Compiler Design in C	Allen Holub	-	-	Lab Sessions
3	Basic of Compiler Construction	Torben Mogensen	-	-	5,6

4.d

Reading latest / top rated research papers (at least 5 papers)

Name of Paper	Name of Authors (Background)	Published in		Problem Statement
		Date	Journal	
Compiler-Assisted Compaction/Restoration of SIMD Instructions	Juan M. Cebrian, Thibaud Balem, Adrian Barredo, Marc Casas, Miquel Moreto, Alberto Ros, Alexandra Jimborean	April 2022	IEEE Transactions On Parallel And Distributed Systems, Vol. 33, No. 4	This paper proposes a novel approach to improve execution efficiency in predicated vector codes, the Compiler-Assisted Compaction/Restoration (CACR) technique. In CACR, the compiler analyzes the code looking for key information required to configure CR. Then, it passes this information to the processor via new instructions inserted in the code. This prevents CR from waiting for active elements on scenarios when it would fail to form dense instructions.
An LLM Compiler for Parallel Function Calling	Sehoon Kim, Suhong Moon, Ryan Tabrizi, Nicholas Lee, Michael W. Mahoney, Kurt Keutzer, Amir Gholami	December 2023	arXiv preprint arXiv:2312.04511	This paper introduces LLMCompiler, which executes functions in parallel to efficiently orchestrate multi-function calling. LLMCompiler streamlines parallel function calling in LLMs with three components: (i) an LLM Planner, formulating execution strategies and dependencies; (ii) a Task Fetching Unit, dispatching and updating function calling tasks; and (iii) an Executor, executing these tasks in parallel. With LLMCompiler, the user specifies the tools along with optional in-context examples, and LLMCompiler automatically computes an optimized orchestration for the function calls. LLMCompiler can be used with open-source models such as LLaMA-2, as well as OpenAI's GPT models.
Parglare: A LR/GLR parser for Python	Igor Dejanović (Faculty of Technical Sciences,	Feb 2022	Science of Computer Programming	Parglare is a Python parsing library that implements deterministic LR and its generalized extension GLR

	University of Novi Sad, Novi Sad, Serbia)			algorithms. Parglare strives to be easy to use by providing rich error messages, visualization, a CLI tool for grammar development, and good documentation. The same grammar format is used for both algorithms. It is easy to choose either LR parsing if performance is more important or GLR in case a grammar cannot fit into the constraints of deterministic LR parsing and a more powerful parsing is needed. Parglare has been used in data extraction from various textual formats, analysis of legacy source code, and developing and teaching DSL development.
Blending Dependency Parsers With Language Models	Nicos Isaak (Computational Cognition Lab, Cyprus)	2023	In Proceedings of the 15th International Conference on Agents and Artificial Intelligence (ICAART 2023) - Volume 3, pages 813-820	To address this complex problem of achieving better results with language models when training data are limited, the authors present a simplified way that automatically improves their learned representations with extra-linguistic knowledge. Further fine-tuning with semantics from state-of-the-art dependency parsers improves existing language models on specialized downstream tasks.
Active Learning for Multilingual Semantic Parser	Zhuang Li, Gholamreza Haffari (Openstream.AI)	2023	arXiv preprint arXiv:2301.12920	Current multilingual semantic parsing (MSP) datasets are almost all collected by translating the utterances in the existing datasets from the resource-rich language to the target language. However, manual translation is costly. To reduce the translation effort, this paper proposes the first active learning procedure for MSP (AL-MSP). AL-MSP selects only a subset from the existing datasets to be translated. The authors also propose a novel selection method that prioritizes the examples diversifying the logical form structures with more lexical choices, and a novel hyperparameter tuning method that needs no extra annotation cost.

4.e

Based on research paper an identify the current Problem statement

Problem Statement			Used in				
	Quiz	Assignment	Lab	Mini Project	Poster Presentation	Test	Any Other

Designing Parser for a simple language like calculator			✓				
--	--	--	---	--	--	--	--

4.f

Identify Companies / Industries which use the knowledge of the subject and thus may provide Internships and final Placements

Name of the Company	To be / Contacted for		
	Student Internship	Student Final Placement	Faculty Internship
Sun Microsystems		✓	
Oracle		✓	

4.g

Identify suitable relevant TOP Guest Speakers from Industry (CS50 Lecture by Mark Zuckerberg - 7 December 2005 - YouTube)

Name of the Identified Guest Speaker	Designation	Name of the Company
Ravishankar Kolachana	Director	Qualcomm Innovation Center Inc

4.h

Identify relevant Technical competitions to participate [Competitions -Paper Presentations, Projects, Hackathons, IVs etc..]

Name of the Relevant Technical Competition Identified to participate	Organized by	Date of the Event
Languages, Compilers, Tools and Theory of Embedded Systems LCTES 2024	ACM SIGPLAN/SIGBED	Mon 24 - Fri 28 June 2024

4.i

Identify faculty in TOP schools / Universities who are teaching same / similar subject and develop rapport e.g. Exchange Lecture Material (Assignments / Tests / Project etc..), Joint Paper Publication

University	Name of the Course	Name of Faculty	Type of Collaboration		
			Exchange of Lecture Material	Joint Publication/ Research	Other
Cornell University	CS 4120/4121/5120/5121 : Introduction to Compilers	Andrew Myers	✓		

4.j

Module Best Available in - Title best resource [from 4.a to 4.d in this AAP] & give details

Module No.	Title of the Module	Web Link	Mention the Title			
			Journal	E-Journal	Magazine	Other Resource
1	Introduction to System Software	https://academic.oup.com/comjnl https://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=12	✓			

2	Assemblers	https://academic.oup.com/comjnl https://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=12	✓			
3	Macros and Macro Processor	https://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=12	✓			
4	Loaders and Linkers	https://academic.oup.com/comjnl https://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=12	✓			
5	Compilers: Analysis Phase	https://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=12	✓			
6	Compilers: Synthesis phase	https://academic.oup.com/comjnl https://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=12	✓			

4.k Referred to any top-rated university in that subject for content

University	Name of the Course	Name of Faculty	Date of Delivery of the Course	Remarks
University of Washington	CSE 401/M501 Compiler Construction	Hal Perkins	Autumn 2023	

4.l Faculty received any certification related to this subject. List of Certifications Identified / Done

Course	Certifying Agency	No. of Hours	Level of the Course		Certification		Remarks
			Introductory	Advance Skill Development	Done on	Proposed to be on	
Compiler Design	NPTEL - Swayam	12 weeks	✓			May 2024	

4.m Completed subject wise/cluster wise training with cluster mentor. List of relevant Refresher Course Identified / Done

Course	Certifying Agency (As suggested by DAB/Cluster Mentor/Industry/ University other than MU)	Certification		Remarks
		Done on	Proposed to be on	
Pedagogy	Udemy	May 2020	-	Outcome Based Education & Academic Quality Assurance
Sub. Content Training	NPTEL -Swayam	-	May 2024	Compiler Design

4.n Best Practices Identified and adopted

No.	Item	Best Practices Identified		
		Univ. 1	Univ. 2	Univ. 3

1	Microsite	https://www.cse.iitk.ac.in/users/swarnendu/courses/spring2023-cs335/	http://www.cse.iitm.ac.in/~kris hna/cs3300/	https://cse.iitkgp.ac.in/~bivasm/compiler2013.html
2	Video Lectures		✓	
3	Assignments		✓	✓
4	Mini Project			
5	Assessment Metric			
6	Quizzes		✓	
7	Labs/ Practical (PBL)			✓
8	Tests			
9	Peer Assessment			
10	Any Other	Slides		

4.o Web Links for Online Notes/YouTube/VIT Digital Content/VIT Lecture Capture/NPTEL Videos

Students can view lectures by VIT professors, captured through LMS 'Lecture Capture' in VIT campus for previous years.

No.	Websites / Links	Module Nos.
1	http://dragonbook.stanford.edu/lecture-notes/Stanford-CS143/03-Lexical-Analysis.pdf	5
2	https://lambda.uta.edu/cse5317/notes/node6.html	5
3	http://www.montefiore.ulg.ac.be/~geurts/Cours/compil/2011/03-syntaxanalysis-3.pdf	5

4.p Recommended MOOC Courses like Coursera / NPTEL / MIT-OCW / edX/VAC etc.

Sr. No.	MOOC Course Link	Course conducted by – Person / University / Institute / Industry	Course Duration	Certificate (Y / N)
1	https://ocw.mit.edu/courses/6-004-computation-structures-spring-2017/pages/c11/	MIT - Massachusetts Institute of Technology		
2	https://www.edx.org/course/compiler	Stanford University		Y
3	https://onlinecourses.nptel.ac.in/noc21_cs07/preview	By Prof. Santanu Chattopadhyay, IIT Kharagpur	12 WEEKS	Y

5 Consolidated Course Lesson Plan

	From (date/month/year)	From (date/month/year)	Total Number of Weeks
Semester Duration	15/01/2024	27/04/2024	15

Week	Lecture no.	Module No.	Lecture Topics / MSE / BSA planned to be covered	Actual date of Completion (Handwritten)	COs Mapped	Recommended Prior Viewing / Reading	
						Lecture No. (on LMS)	Chapter No./ Books/ Web Site
1	1,2	1	Concept of System Software, Goals of system software's, system program and system programming, Introduction to various system programs such as Assembler, Macro processor, Loader, Linker, Compiler, Interpreter, Device Drivers, Operating system, Editors, Debuggers		1		T3,Chp#1 Pg-4-7
	3	2	Introduction to various system programs such as Assembler, Macro processor, Loader, Linker, Compiler, Interpreter, Device Drivers, Operating system, Editors, Debuggers		2		T3 Chp#3 Pg- 60
2	4,5	2	Elements of Assembly Language programming, Assembly scheme, pass structure of assembler		2		T3 Chp#3 Pg- 60
	6	2	Assembler Design: Two pass assembler Design		2		T3 Chp#3 Pg- 60
3	7,8	2	Assembler Design: Single pass assembler Design		2		T3 Chp#3 Pg- 60
	9	3	Design for Hypothetical / X86 family processor, data structures used.		3		T3 Chp#3Pg- 62-77
4	10,11	3	Introduction, Macro definition and call, Features of Macro facility: Simple, parameterized, conditional and nested.		3		T3 Chp#3Pg- 62-77
	12	3	Design of single pass macro processor, data structures used.		3		T3 Chp#3Pg- 62-77
5	13,14		Design of Two pass macro processor, data structures used.		3		T3 Chp#3Pg- 62-77

Week	Lecture no.	Module No.	Lecture Topics / MSE / BSA planned to be covered	Actual date of Completion (Handwritten)	COs Mapped	Recommended Prior Viewing / Reading	
						Lecture No. (on LMS)	Chapter No./ Books/ Web Site
	15	4	Introduction, functions of loaders, Relocation and Linking concept,		4		T1 Chp#3Pg-82-94
6	16,17	4	Different loading schemes: Relocating loader,		4		T1 Chp#3Pg-82-94
	18	4	Absolute Loader & DLL,		4		T1 Chp#3Pg-82-94
7	19,20	4	Direct Linking Loader, Dynamic linking and loading		4		T1 Chp#3Pg-82-94
	21	5	Introduction to compilers, Phases of compilers: Lexical Analysis- Role of Finite State Automata in Lexical Analysis,		5		T2 Chp#4 Pg-122-136 T2 Chp#5 Pg-150-167
8	22,23	5	Design of Lexical analyser, data structures used		5		T2 Chp#4 Pg-122-136
	24	5	Syntax Analysis- Role of Context Free Grammar in Syntax analysis,		5		T2 Chp#5 Pg-150-167
9	25,26	5	Types of Parsers: Top down parser- LL(1),		5		T2 Chp#4 Pg-122-136
	27	5	Bottom up parser- Operator precedence parser, SLR		5		T2 Chp#5 Pg-150-167
10	28,29	5	Semantic Analysis, Syntax directed definitions.		5		T2 Chp#4 Pg-122-136 T2 Chp#5 Pg-150-167
	30	6	Intermediate Code Generation: Types of Intermediate codes: Syntax tree		6		T2 Chp#5 Pg-168-178

Week	Lecture no.	Module No.	Lecture Topics / MSE / BSA planned to be covered	Actual date of Completion (Handwritten)	COs Mapped	Recommended Prior Viewing / Reading	
						Lecture No. (on LMS)	Chapter No./ Books/ Web Site
11	31,32	6	Postfix notation,		6		T2 Chp#1 Pg- 20-22
	33	6	Three address codes: Triples and Quadruples.		6		T2 Chp#1 Pg-1-22
12	34,35	6	Code Optimization: Need and sources of optimization,		6		T2 Chp#3 Pg- 84-134
	36	6	Code optimization techniques: Machine Dependent and Machine Independent.		6		T2 Chp#4 Pg- 181-190
13	37,38	6	Code Generation: Issues in the design of code generator, code generation algorithm.		6		T2 Chp#4 Pg- 195-202
	39	6	Basic block and flow graph		6		R1 Chp#3 Pg- 30-36

6

Rubric for Grading and Marking of Term Work (inform students at the beginning of semester)

Class Participation	Theory (ISA= 20)				Practical (ISA= 25)				Total
	Quiz (Best of 2)	Take Home Test (Best of 2)	Poster Presentation	Other Assignments	Lab Participation	Lab work	Mini Project (Implementation + Seminar)	PB L	
05	03	03	03	06	05	10	05+02 = 07	03	45

7

Assignments / Tutorials Details

Assignment/ Tutorial No.	Title of the Assignments / Tutorials	CO Map	Assignment/ Tutorials given to Students on	Week of Submission
--------------------------	--------------------------------------	--------	--	--------------------

1	Quiz on System Software and basics of assembly language	CO1	Week 2	Week 2
2	Data Structure with entries of 2 pass assembler for a given ALP	CO2	Week 3	Week 4
3	Take Home Test – Converting given ALP to Machine Language	CO2	Week 4	Week 5
4	Minute Paper - Macro	CO3	Week 5	Week 6
5	Data structure for Nested Macro	CO3	Week 6	Week 7
6	Designing Macro-processor with mentioned features	CO3	Week 7	Week 8
7	Poster on different Linking and Loading schemes	CO4	Week 8	Week 9
8	Quiz on Compiler phases, RE and CFG	CO5	Week 9	Week 9
9	Assignment on LA and Top-down parser	CO5	Week 10	Week 11
10	Take Home Test – Designing bottom-up parser for the given grammar	CO5	Week 11	Week 12
11	Generating intermediate code for a given statement/block of code	CO6	Week 13	Week 14
12	Optimization of the given block of code	CO6	Week 14	Week 15

Analysis of Assignment / Tutorial Questions and Related Resources

Assignment / Tutorial No.	Week No.	Type* (√)			Module No.	Based on #			Question Type (√)	
		R	PQ	OBT		Text Book	Reference Book	Other Learning Resource	MU EQ	Thought Provoking
1	Week 2		✓			1, 3	4			
2	Week 3	✓				1, 3	4			
3	Week 4			✓		1, 3	4			
4	Week 5	✓				1, 3	4			
5	Week 6	✓				1, 3	4			
6	Week 7					1, 3	4			✓
7	Week 8	✓				1, 3	4			

8	Week 9		✓			2	1, 2, 3			
9	Week 10	✓				2	1, 2, 3			
10	Week 11			✓		2	1, 2, 3			
11	Week 13					2	1, 2, 3			✓
12	Week 14					2	1, 2, 3			✓

* Tick (✓) the Type of the Assignment: Regular (R); Pop Quiz (PQ) ; Open Book Test for TE/BE/ME (OBT)

Write number for text book, reference book, other learning resource from this AAP – from Points 4.a to 4.d

8

Internal Assessment / Other Class Test / Open Book Test (OBT)/Take Home Test (THT) Details

Tests	Test Dates	Module No.	CO Map	Question Paper Pattern	Policy
MSE	3 Mondays of the semester (MSE-1, MSE-2, MSE-3)	MSE-1 – 1, 2 MSE-2- 3, 4 MSE-3 – 5	1, 2, 3, 4, 5	Q1 – 8 Sub Questions of 2 marks each, Attempt any 5 Q2 & Q3 – Descriptive & will be of <ul style="list-style-type: none"> 3 Sub questions of 5 marks each & student will attempt any 2 Or 2 Sub questions of 10 marks each & student will attempt any 1 	Top 2 scores of the 3 MSEs will be considered towards 30% of total evaluation of the course. No Repeat MSE.
Pop Quiz	Week 2 and Week 9	1, 5	1, 5	MCQs	Best of 2 will be considered for evaluation
Take Home Test	Week 4 and Week 11	2, 5	2, 5	Descriptive questions	Best of 2 will be considered for evaluation

9.a

Practical Activities

Practical No.	Module No.	Title of the Experiments	Type of Experiment		Topics to be highlighted	CO Map
			PBL	Newly Added		
1	2	To develop and implement Pass1 of 2 Pass Assembler for IBM 360/370			Assembler	CO2
2	2	To develop and implement Pass2of 2 Pass Assembler for IBM 360/370			Assembler	CO2
3	3	2 Pass Macro Pre-Processor with nested macro calls		✓	Macro-processor	CO3

4	4	Linker: Write a program in C and in ALP and link the object file of both the programs using linker (ELF/PIC)			Lexical Analyser	CO4
5	5	Lexical Analyser for expression using operators (+, -, *, /, ^) and literals (integer and float)			Top-Down Parser	CO5
6	5	Predictive parser for a given specific grammar			Parser	CO5
7	5	Operator Precedence Parser for calculator	✓		Parser	CO5
8	5	Construct SLR parser for the given grammar.		✓	Parser	CO5
9	6	Implementation of Intermediate code generation phase of compiler.			ICG	CO6
10	6	Implementation of code generation phase of compiler.			Code Generation	CO6
11	5, 6	Case Study on LEX & YACC			Tools	CO5, CO6

10

Beyond Syllabus Activities for Gap Mitigation

No.	Type of the Activity	Activities	Number of beneficiaries	Other Details – guest profile, feedback, mark sheet, report
1	Experiential learning/Interaction with Outside World	1- Guest Lectures by Industry Expert	NA	
		2- Workshops	NA	
		3- Mini Project	Yes	
		4- Industrial Visit	NA	
		5- Any other activity	NA	
2	Collaborative & Group Activity	6- Poster Presentation	Yes	
		7- Minute Papers	Yes	
		8- Students Seminars	Yes	
		9- Students Debates	NA	
		10- Panel Discussion / Mock GD	NA	
		11- Mock Interview	NA	

		12- Any other activity	NA	
3	Co-Curricular Activity	13- Informative videos (NPTEL/Youtube /TEDx/ MIT OW/edX)	Yes	
		14- Lecture Capture Usage	Yes	
		15- Any other activity	NA	
4	Tests & Assessments	16- Class Tests/ Weekly Tests	NA	
		17- Pop Quiz	Yes	
		18- Mobile App Based Quiz	NA	
		19- Open Book Test	NA	
		20- Take Home Test	Yes	
		21- Any other activity	NA	

*** Do not delete any activity. Give details for planned events. Write 'NA' for activity Not Planned.**

Consolidated Academic Administration Plan Prepared by (mention all theory teaching faculty names with signature)

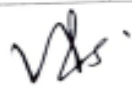
Please write below your name and sign with date of the external cluster mentor meeting



Pankaj Vanwari

Dr. Jayant Dani
External Industry Mentor

Dr. Manish Potey
External Academic Mentor



Pankaj Vanwari
VIT Cluster Mentor



Dr. Mandar Sohani
Program HOD