School of Technology, Pandit Deendayal Energy University, Gandhinagar

Course File (A to Z Essentials)

Name of the Course:	System Software & Compiler Design
Course Code:	20CP302T
Program:	B.Tech
Department:	Computer Science and Engineering
Semester:	V
Academic Year:	Odd (2023-2024)
Name of Course Coordinator:	Dr. Shivangi Surati
Names of the Other Faculty Members:	Dr. Shivangi Surati, Dr. Rajiv Gupta, Dr. Meera Khanna

A.	Course Syllabus, Pre requisites for the Course
B.	Teaching Schemes
C.	Course Outcomes (COs)
D.	Mapping of Course Outcomes with Programme Outcomes (PO)
E.	Academic Calendar and Class Time Table
F.	Lesson Plan
G.	Evaluation Scheme and Rubrics
H.	List of Books and Reference books
I.	Class Notes, Handouts, Presentations etc.
J.	Tutorials, Assignments, Case Studies, Quiz, etc.
K.	ICT - Course related Web-links, Software, E-books, Relevant NPTEL and MOOC, Video Lectures, Blogs, Virtual Lab, Animation, Simulation, etc.
L.	Laboratory Manuals (if applicable)
M.	List of International / National Journals related to the Course (if applicable)
N.	List of Classic Journal Papers / Articles / Review Papers related to the Course
O.	List of world leading Industries / Organizations / working on the course related areas
P.	List of world leading Scientists / Academicians working on the course related areas
Q.	Copies of the Mid and End Semester Examination Question Papers (Past)
R.	Attendance Record
S.	Records of the Continuous Assessment (Assignment, Quiz, Laboratory Work, etc.)
T.	Details of Remedial Classes (with evidences)
U.	Details of Expert Lectures / Industrial Visits/Events (Only related to the course)
V.	List and Slow and Advanced Learners, activity planned and executed
W.	Direct Assessment (Result of mid, end and internal assessment components)
X.	Indirect Assessment (Exit Survey/Post Test)
Y.	Final Attainment of COs and POs and Interpretation
Z.	Actions to be taken if COs and POs are not achieved

Date:

Signature of Subject Teachers **Signature of Department Coordinator (IQAC)** **Signature of Head of the Department**

A. Course Syllabus, Pre requisites for the Course

Course Syllabus:

		Mapped CO
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.**	**CO1**
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.**	**CO2**
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.**	**CO3, CO4, CO6**
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.**	**CO5**
	Max. 39 Hrs.	

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

B. 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

COURSE OBJECTIVES:

- Define and learn system Software such as Assemblers, Loaders, Linkers, macro-preprocessors.
- Familiarize with source file, object file and executable file structures and libraries.
- Describe the front-end and back-end phases of compiler and their importance to students.
- Learn Lexical Analysis, Syntax Analysis and Semantic Analysis.
- Learn to generate Intermediate Code and code optimization.

C. 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.

D. 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)

E. Academic Calendar and Class Time Table

ACADEMIC CALENDAR 2023-24 (Odd SEMESTER)

Odd Semester: UG Sem.1/3/5/7 & PG Sem. 1/3 (FoET) & UG Sem. 1/3/5/7 & PG Sem 1/3 (FoLS)	
Particulars	Date
Semester Registration & Commencement of classes-FoET & FoLS- 1st Sem	17th July (Mon) 2023
Semester Registration, Department Orientation & Commencement of classes for 3/5/7 Sem – FoET & FoLS	24th Jul (Mon). 2023
Evaluation of Rural Internship/CSSI & Evaluation of Industry Orientation, & Evaluation of Industrial Internship	7th (Mon)-11th (Fri)Aug. 2023
Independence Day Celebration	15th Aug. (Thes) 2023
Attendance Review-1 (After 4 week)	17th (Thur)-18th (Fri) Aug. 2023
Internal Assesment-1 (Quiz, Test, Assignment etc.)** Student mentoring week – 1	21st (Mon)-25th (Fri)Aug. 2023
Mid Semester Examination / Project Phase 1 Review	11th Sept. (Mon) 2023 Onwards
Attendance Review-2 (After 8 week)	14th (Thur)-15th (Fri)Sept 2023
Parent Teacher Meeting (Saturday)	23rd Sept.(Sat) 2023
Last date of showing evaluated answer books of Mid Semester Examination	27th Sept. (Wed) 2023
Declaration of Mid Semester Exam Result	6th Oct. (Fri) 2023
360 Degree Feedback from Students by School Admin	9th (Mon)-13th (Fri)Oct. 2023
Attendance Review-3 (After 12 week)	12th (Thur)-13th (Fri)Oct 2023
Rangtaal – Navratri Celebration	13th Oct.(Fri) 2023
Internal Assesment-2 (Quiz, Test, Assignment etc)** Student mentoring week – 2	25th (Wed)-31st (Tues)Oct. 2023
Tesseract – The Science & Technical Fest	03(Fri)-04(Sat)-05(Sun) Nov. 2023
Declaration of Detention list of students (during 13th Week)	By 20th Oct (Fri) 2023
Diwali Vacation	13th (Mon)-17th (Fri) Nov. 2023
Classes End	21st (Tues) Nov. 2023
Practical Examinations, submission of Term Work and Seminars	22nd Nov.(Wed) 2023 Onwards
Dissertation presentation for UG and PG for FOLS	22nd Nov.(Wed) 2023 onwards
End Semester Examinations - FoET& FoLS	28th Nov.(Tues) 2023 Onwards
Last date of Submission of Marks of End sem. Exam	15th Dec. (Fri) 2023
Rural Internship for FoLS students	During Dec 2023
Project Phase I Exam for PG program of FoET & Progress Review for Ph. D.	18th (Mon)-22nd (Fri)Dec. 2023
Winter Break	26th (Tues)-29th (Fri)Dec. 2023
Alumni Day	29th Dec (Fri) 2023
Even Semester: UG Sem. 2/4/6/8 & PG Sem. 2/4 (FoET) & UG Sem.2/4/6/8 & PG Sem. 2/4 (FoLS)	
Next semester registration	27th (Wed)-30th (Sat) Dec. 2023
Start of Next Semester	1st Jan. (Mon) 2024

Class Time Table and Faculty Time Table with office hours

Class Time-table (Semester-5, Div-3)

Pandit Deendayal Energy University
School of Technology

Autumn Semester 2023

Day	09:00-10:00	10:00-11:00	11:00-12:00	13:00-14:00	14:00-15:00	15:00-16:00	16:00-17:00	17:00-18:00
Monday	G7(B6)8 (20CP301T) D008, SOVY-L G1(2) (20CP305T) F-504, PCH-L G4(5)G6 (20CP301T) E203, VIMI-L G1(2)G3 (20CP301T) E304, RUZ-L G1(2) (20CP302T) F-401, SVS-L		G5G6 (20CP301T) F-504, MAPL-L	G (OE3) F-404, OE4F-L	G6 (20CP301P) E112-113, KSI-P G5 (20CP301P) E112-113, MAPL-P		G1 (20CP302P) F-104, SVS-P G1 (20CP301P) F-203, RUZ-P G4 (20CP301P) F-401, VIMI-P G5G3 (20CP305P) E112-113, PCH-P G7 (20CP301P) E216, SOVY-P	
Tuesday	G1G2 (20CP302T) F-401, SVS-L G1G2G3 (20CP301T) E204, RUZ-L G4G5G6 (20CP301T) E203, VIMI-L G3G1 (20CP305T) F-504, PCH-L G7(B6)9 (20CP301T) D007, SOVY-L	G5G6 (20CP302T) F-504, SVS-L	G5G6 (20CP301T) F-504, MAPL-L	G (OE3) F-404, OE4F-L	G5G6 (CDC) F-402, CDCT-L	G5G6 (20CP304T) F-404, HAK-L	G5 (20CP306P) F-202, KMS-P G6 (20CP306P) F-103, ABL-P	
Wednesday	G5G6 (20CP302T) F-504, SVS-L	G5G6 (20CP303T) F-504, KBT-L	G5G6 (20CP302T) F-504, KBT-L	G (OE3) F-404, OE4F-L	G5G6 (20CP301T) F-504, MAPL-L	G5G6 (20CP304T) E005, HAK-L	G5 (20CP304P) E216, HAK-P G6 (20CP302P) F-504, SVS-P	
Thursday	G5 (20CP301P) F-202, VIMI-P G8 (20CP301P) E215, SOVY-P G2 (20CP301P) F-103, RUZ-P		G5G6 (20CP302T) F-504, KBT-L	G5G6 (20F301T) E004, ISX-L G5G6 (20F301T) E004, HI-E-L G5G6 (20F301T) E004, HI-M-L G5G6 (20F101T) E004, HV1-L	G6 (20CP306P) F-203, RISC-P G5 (20CP302P) F-401, SVS-P		G5G6 (20F202P) F-PDH-SE, CHJH-P G5G6 (20F202P) F-MMI-SE, MBK-P G5G6 (20F201P) C203, GAM-P G5G6 (20F201P) C206, MNP-P	
Friday	G6 (20CP305P) E213, VIMI-P G9 (20CP301P) E216, SOVY-P G3 (20CP301P) F-103, RUZ-P		G5G6 (20CP302T) F-504, SVS-L	G5G6 (20F301T) E004, ISX-L G5G6 (20F301T) E004, HI-E-L G5G6 (20F301T) E004, HI-M-L G5G6 (20F101T) E004, HV1-L	G5G6 (20CP306T) F-403, RIMO-L	G5G6 (CDC) F-403, CDCT-L		

Faculty Abbr.	Faculty Name
ABL	Ashka Raval
CDCT	CD Cell Trainer
CHJH	Chandan Kumar Jha (Visiting Fac)
GAM	Garima Mishra
HAK	Hargeet Kaur
HI-E	Industrial 4.0 Electrical Faculty
HI-M	Industrial 4.0 Mechanical Faculty
HV1	Industry4.0 Faculty_ICT_1
KBT	Krishna Brahmabhatt
KMS	Komal Singh
KSL	Ketan Sable
MAPL	Manish Pathak
MBK	M B KIRAN
MNP	Manan Pathak-VT
ODEF	OE Fac
PCH	Payal Chaudhari
PCH	Payal Chaudhary
RIMO	Ritiksha Modi (ADF)
RISC	Rinkle Jain
RSX	Rajat Saxena
RUZ	Ruzvil H Jhaveri
SOVY	Soham Vyas
SVS	Shivangi Surati
VIMI	Vipul Mishra

Subject Abbr.	Subject Name
20CP301T	Computer Network
20CP302P	System Software & Compiler Design - Lab
20CP302T	System Software & Compiler Design
20CP303T	Software Engineering
20CP304T	Information Security
20CP301P	Computer Network - Lab
20CP304P	Information Security - Lab.
20IF301P	Industry 4.0 Lab
20IF301T	Industry 4.0
23CP301P	Advanced Python Programming - Lab.
23CP301T	Advanced Python Programming
23CP302P	Computer Graphics - Lab.
23CP302T	Computer Graphics
23CP305P	Advanced Java - Lab
23CP305T	Advanced Java
23CP306P	Introduction to Web Technology - Lab
23CP306T	Introduction to Web Technology
CDC	CDC
OE3	OE Sem 5

Dr. Santosh Bharti
Timetable Coordinator

Dr. Shakti Mishra
HoD

Prof. Dhaval Pujara
Director
(School of Technology)

Class Time-table (Semester-5, Div-6)

Pandit Deendayal Energy University
School of Technology
B.Tech - Computer Engineering

Autumn Semester 2023

Day	09:00-10:00	10:00-11:00	11:00-12:00	12:00-13:00	13:00-14:00	14:00-15:00	15:00-16:00	16:00-17:00	17:00-18:00
Monday	G466566 [20CP401P] E204, VIMH-L		G11612 [20CP402T] D009, SVS-L		G [063] F-404, DMH-L	G12 [20CP404P] F-401, HAK-P		G162 [20CP402P] F-304, SVS-P	
	G162563 [20CP401P] E204, R12-L					G11 [20CP402P] F-304, DHJ-P		G162 [20CP405P] E112-114, PCH-P	
	G76d969 [20CP401P] D008, SOYY-L							G7 [20CP401P] E216, SOYY-P	
	G162 [20CP405T] F-506, PCH-L							G1 [20CP401P] F-304, R12-P	
	G162[20CP402T] F-401, SVS-L							G4 [20CP401P] F-401, VIMI-P	
Tuesday	G76d969 [20CP401P] D007, SOYY-L		G11612 [20CP402T] F-402, TUKA-L		G [063] F-404, DMH-L	G12 [20CP406P] F-202, RJIC-P		G11612 [20CP401T] F-402, DHJ-L	G11612 [20CP404T] F-304, R12-L
	G162 [20CP402T] F-401, SVS-L					G11 [20CP404P] F-304, R12-P			
	G162 [20CP405T] F-506, PCH-L								
	G162563 [20CP401P] E204, R12-L								
	G466566[20CP401P] E216, VIMI-L								
Wednesday	G11612 [20CP402T] D009, TUKA-L	G11612 [20CP404T] D009, TUKA-L	G11612 [20CP405T] D009, RMS-L		G [063] F-404, DMH-L	G11612 [20CP402T] E204, SVS-L	G11612 [20CP402T] E203, SVS-L	G9610 [CDC] F-402, CDCT-L	G9610 [CDC] E002, CDCT-L
Thursday	G8 [20CP401P] E213, SOYY-P			G11612 [20IF201T] E204, SMIT-L		G12 [20CP401P] E112-114, KSL-P		G11612 [20CP401T] D010, DHJ-L	G11612 [20CP401T] D010, DHJ-L
	G2 [20CP401P] F-304, R12-P			G11612 [20IF201T] E204, H-H-L		G11 [20CP401P] E112-114, NIPA-P			
	G5 [20CP405P] F-202, VIMI-P			G11612 [20IF201T] E204, H-MI-L					
				G11612 [20IF201T] E204, HJD-L					
Friday	G6 [20CP401P] E214, VIMI-P	G11612 [20CP404T] D102, R12-L	G11612 [20IF201T] E204, SMIT-L		G12 [20CP402P] E004, SVS-P		G11612 [20IF201P] F-MMS-SR, MNP-P		
	G3 [20CP401P] F-203, R12-P		G11612 [20IF201T] E204, H-H-L		G11 [20CP406P] F-213, DHJ-P		G11612 [20IF201P] E112-114, GAM-P		
	G9 [20CP401P] E216, SOYY-P		G11612 [20IF201T] E204, H-MI-L				G11612 [20IF201P] F-PCH-SR, CHDH-P		
			G11612 [20IF201T] E204, HJD-L				G11612 [20IF201P] C004, MNP-P		

Faculty Abbr.	Faculty Name
CDCT	CD Cell Trainer
CHB	Chintan Bhatt
CHDH	Chandan Kumar Jha [Visiting Fac]
DHJ	Dhara Joshi (ADF)
GAM	Garima Mishra
HAK	Hargeet Kaur
HI-E	Industrial 4.0 Electrical Faculty
HI2	Industry4.0 Faculty_ICT_2
HI-M	Industrial 4.0 Mechanical Faculty
KMS	Komal Singh
KSL	Ketan Sable
MBK	M B KIRAN
MNP	Manan Pathak-VF
NIPA	Nitin Padaria (VF)
OE-IF	OE Fac
PCH	Payal Chaudhari
RJSC	Rinkle Jain
RUZ	Rutvij H Jhaveri
SMIT	Shirishendu Mitra
SOYY	Soham Vyas
SVS	Shivangi Surati
TUKA	Tushar Kakaiya (VF)
VIMI	Vipul Mishra

Subject Abbr.	Subject Name
20CP201T	Computer Network
20CP202P	System Software & Compiler Design - Lab
20CP202T	System Software & Compiler Design
20CP203T	Software Engineering
20CP204T	Information Security
20CP201P	Computer Network - Lab
20CP204P	Information Security - Lab.
20IF201P	Industry 4.0 Lab
20IF201T	Industry 4.0
23CP201P	Advanced Python Programming - Lab.
23CP202P	Computer Graphics - Lab.
23CP202T	Computer Graphics
23CP205P	Advanced Java - Lab
23CP205T	Advanced Java
23CP206P	Introduction to Web Technology - Lab
23CP206T	Introduction to Web Technology
CDC	CDC
OE3	OE Sem 5

Dr. Santosh Bharti
Timetable Coordinator

Dr. Shakti Mishra
HoD

Prof. Dhaval Pujara
Director
(School of Technology)

Faculty Time Table

Shivangi Surati

Day	09:00-10:00	10:00 to 11:00	11:00 to 12:00	12:00 to 13:00	13:00 to 14:00	14:00 to 15:00	15:00 to 16:00	16:00-17:00	17:00-18:00
MON	CG (23CP302T) – all div F401, CP(5) - L		G11G12 (20CP302T) D-009, CP(5) - L					CG (23CP302P) – all div F-104, CP(5) - P	
TUE	CG (23CP302T) – all div F401, CP(5) - L	G5G6 (20CP302T) F-504, CP(5) - L							
WED	G5G6 (20CP302T) F-504, CP(5) - L					G11G12 (20CP302T) E-203, CP(5) - L	G11G12 (20CP302T) E-203, CP(5) - L	G6 (20CP302P) F-504, CP(5) - P	
THURS						G5 (20CP302P) F-401, CP(5) - P			
FRI			G5G6 (20CP302T) F-504, CP(5) - L			G12 (20CP302P) E004, CP(5) - P			
SAT									

Office hour: Friday, 12:00 pm to 2:00 pm

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

Division-3

Lecture No	Topic	Tentative Date	Actual date	Remarks
Unit-1 (BW + PPT)				
1	Introduction to language processors	25/7/23	25/7/23	
2	Introduction to different phases of compiler	26/7/23	26/7/23	
3	Symbol table and error handling	28/7/23	28/7/23	
4	Alphabets And Tokens In Computer Languages, Representation, Token Recognition	1/8/23	1/8/23	
5	Introduction to finite automata- NFA and DFA and conversion from NFA to DFA	2/8/23	2/8/23	
6	RE to DFA using syntax tree method	4/8/23	4/8/23	
7	Lexical errors, error recovery, Input buffering	8/8/23	8/8/23	
Unit-2 (BW + PPT)				
1	Syntax Analysis- Introduction, role of parsers, lexer-parser communication	9/8/23	9/8/23	
2	Context Free Grammars, ambiguity	11/8/23	11/8/23	
3	Elimination of left recursion, left factoring	16/8/23		
4	Introduction of Top-down parser, Recursive Descent Parser, Introduction of Predictive parser	18/8/23		
5	Predictive parser –first and follow	22/8/23		
6	Predictive parser - Table and algorithm	23/8/23		
7	Introduction of Bottom up parser, handle, handle pruning	25/8/23		
8	Operator precedence parser table	29/8/23		
9	Operator precedence parser algorithm, error recovery	1/9/23		
10	LR parser- SLR parsing items generation	5/9/23		
11	SLR parsing table	6/9/23		
12	SLR parsing algorithm	8/9/23		
13	CLR parsing items generation			
14	CLR parsing table			
15	LALR parsing			
16	Semantic Analysis			
17	Syntax direct translation			
18	Comparison between SDD and SDT			
Unit-3 (BW + PPT)				
1				
2				
3				
4				
5				
6				
7				

8				
Unit-4 (BW + PPT)				
1				
2				
3				
4				
5				
6				

BW- Board Work, PPT- Powerpoint Presentation

Division-6

Lecture No	Topic	Tentative Date	Actual date	Remarks
Unit-1 (BW + PPT)				
1	Introduction to language processors	24/7/23	24/7/23	
2	Introduction to different phases of compiler	26/7/23	31/7/23	Orientation on 26/7/23
3	Symbol table and error handling	26/7/23	2/8/23	Orientation on 26/7/23
4	Alphabets And Tokens In Computer Languages, Representation, Token Recognition	31/7/23	2/8/23	
5	Introduction to finite automata- NFA and DFA and conversion from NFA to DFA	2/8/23	7/8/23	
6	RE to DFA using syntax tree method	2/8/23	9/8/23	
7	Lexical errors, error recovery, Input buffering	7/8/23	9/8/23	
Unit-2 (BW + PPT)				
1	Syntax Analysis- Introduction, role of parsers, lexer-parser communication	14/8/23		
2	Context Free Grammars, ambiguity	16/8/23		
3	Elimination of left recursion, left factoring	16/8/23		
4	Introduction of Top-down parser, Recursive Descent Parser, Introduction of Predictive parser	21/8/23		
5	Predictive parser –first and follow	23/8/23		
6	Predictive parser - Table and algorithm	23/8/23		
7	Introduction of Bottom up parser, handle, handle pruning	28/3/23		
8	Operator precedence parser table	4/9/23		
9	Operator precedence parser algorithm, error recovery	6/9/23		
10	LR parser- SLR parsing items generation	6/9/23		
11	SLR parsing table			
12	SLR parsing algorithm			

13	CLR parsing items generation			
14	CLR parsing table			
15	LALR parsing			
16	Semantic Analysis			
17	Syntax direct translation			
18	Comparison between SDD and SDT			
Unit-3 (BW + PPT)				
1				
2				
3				
4				
5				
6				
7				
8				
Unit-4 (BW + PPT)				
1				
2				
3				
4				
5				
6				

G. 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 CO's
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/Project	Analytical/design based questions on syllabus covered	40		
Total 50 marks will be converted into 25 marks for the mid-sem evaluation.					
Direct (Internal Evaluation)	Assignments	For each unit a separate assignment will be prepared (Descriptive and analytical/design based questions)	40	CO1,CO2, CO3,CO4	It fractionally contributes to 25% weightage of Direct Assessment to CO attainment.
	Quiz	At the end of semester Question-answer based	10	CO3,CO4, CO5, CO6	

		evaluation on One-to-one basis			
Total 50 marks will be converted into 25 marks for the continuous assessment.					
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.				

H. List of Books and Reference books

TEXT/REFERENCE BOOKS

1. Alfred V Aho, M S. Lam, R Sethi, Jeffrey D. Ullman. Compilers-Principles, Techniques and Tools, Pearson.
2. D. M. Dhamdhere, System software and operating system, TMH

I. Class Notes, Handouts, Presentations etc.

Lecture

- Topic wise PPTs/examples for all units 1 to 4 will be shared with students for following topics:
  - Unit 1
  - Unit 2
  - Unit 3
  - Unit 4

The other material will be provided during lectures.

J. Tutorials, Assignments, Case Studies, Quiz, etc.

Will be provided at the end of each unit (Total 4 assignments). 1 quiz at the end.

K. ICT - Course related Web-links, Software, E-books, Relevant NPTEL and MOOC, Video

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-ycscs1-compilers

Software:

1. Flex Compiler:

https://gnuwin32.sourceforge.net/packages/flex.htm

2. Bison

https://gnuwin32.sourceforge.net/packages/bison.htm

L. Laboratory Manuals (if applicable)

Will be given separately.

M. List of International / National Journals/Conferences related to the Course (if applicable)

International Journals:

ACM Transactions on Architecture and Code Optimization

Conferences:

CC - International conference on Compiler Construction

PLDI – ACM SIGPLAN Conference on Programming Language Design and Implementation

N. List of Classic Journal Papers / Articles / Review Papers related to the 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

O. List of world leading Industries / Organizations / working on 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.

P. List of world leading Scientists / Academicians working on 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

Q. Copies of the Mid and End Semester Examination Question Papers (Past)

Roll No. _____

Pandit Deendayal Energy University

Re-Mid Semester Examination-November 2022

B. Tech. (Computer Science & Engineering)

Semester - V

Date: 07/11/2022

Course Name: System Software & Compiler Design **Time: 2 hrs**

Course Code: 20CP302T **Max. Marks: 50**

Instructions:

1. Do not write anything other than your roll number on question paper.
2. Assume suitable data wherever essential and mention it clearly.
3. Writing appropriate units, nomenclature, and drawing neat sketches/schematics wherever required is an integral part of the answer.

Ques. No.	Description	Marks	CO Mapped	BL
Q.1	**Answer following Questions.**	10*2=**20**		
	a. Solve this expression showing results after each stage (i.e., preprocessor, compiler, assembler, linker etc.) of compiler: **position=initial+rate*60**. Explain each and every individual stage properly. (Note: Use suitable diagrams)		CO1	L3
	b. Construct LL(1) parsing algorithm using following grammar and this input string: 3+5*7. Show each step of tracing input string using stack. E -> E + T | T T -> T * F | F F -> (E) | int		CO2	L6
Q.2	**Answer following Questions (Any THREE).**	05*3=**15**		
	a. Discuss these various types of tokens in short: Language Processor, Machine Language, Assembly Language, Higher-Level Language, Compiler Phase		CO1	L2
	b. Define these terms: Derivation, Parse Tree, Ambiguity, Associativity of Operators, Precedence of Operators		CO1	L1
	c. List down any five applications of Compiler Technology.		CO1	L1
	d. Discuss whether a following grammar is LL(1) or not. $E \rightarrow T\ E'$ $E' \rightarrow +\ T\ E' \mid \lambda$ $T \rightarrow F\ T'$ $T' \rightarrow *\ F\ T' \mid \lambda$ $F \rightarrow (E) \mid \mathbf{id}$		CO2	L2

Q.3	**Answer following Questions (Any THREE).**	05*3=**15**		
	a. Check whether the given grammar is ambiguous or not. S → AB / C A → aAb / ab B → cBd / cd C → aCd / aDd D → bDc / bc Consider a string generated by the given grammar **aabbccdd.**		CO1	L5
	b. Identify the language from each of the following: (1) {a}{a, b}*{b}{a, b}*{a} (2) (b + ab*a)*ab* (3) a + ba* (4) (a + b)*(aa + bb) (5) a (a + b)* b		CO1	L2
	c. Execute Left-Factoring on following grammar: 1. P → E 2. E → E + T 3. E → T 4. T → ident 5. T → int		CO1	L3
	d. Illustrate Tree Terminology in detail.		CO1	L2

Roll No. _____

Pandit Deendayal Energy University

Mid Semester Examination-September 2022

B. Tech. (Computer Science & Engineering)

Semester - V

Date: 23/09/2022

Course Name: System Software & Compiler Design **Time: 2 hrs**

Course Code: 20CP302T **Max. Marks: 50**

Instructions:

1. Do not write anything other than your roll number on question paper.
2. Assume suitable data wherever essential and mention it clearly.
3. Writing appropriate units, nomenclature, and drawing neat sketches/schematics wherever required is an integral part of the answer.

Ques. No.	Description	Marks	CO Mapped	BL
Q.1	**Answer following Questions.**	10*2=**20**		
	a. Solve following expression showing results after each stage (i.e. preprocessor, compiler, assembler, linker etc.) of compiler: ***height = (width+56)*factor(foo)*** (Note: Use suitable diagrams)		CO1	L3
	b. Construct LL(1) parsing algorithm using following grammar and this input string: ***acbbac***. Calculate FIRST and FOLLOW. Show each step of tracing input string using stack. S'->S$ S->aAS|c A->ba|SB B->bA|S		CO2	L6
Q.2	**Answer following Questions (Any THREE).**	05*3=**15**		
	a. Discuss these various types of tokens in short: Keywords, Identifiers, Numbers, Strings, Comments		CO1	L2
	b. Define these terms: Terminal, Non-terminal, Sentence, Context-free Grammar, Start symbol		CO1	L1
	c. List down any five applications of system software.		CO1	L1
	d. Discuss whether a following grammar is LL(1) or not. S -> iEtSS'|a S' ->eS|ε E -> b		CO2	L2

Q.3	**Answer following Questions (Any THREE).**	05*3=**15**		
	a. Consider following grammar: P->E E->E+E E->ident E->int Give your critiques if above grammar is *ambiguous* or not, considering the input as: ***ident + int + int***		CO1	L5
	b. Identify the language from each of the following: 1. (0|1)*.0 2. 0(0+1)*1 3. (a|b)*aa(a|b)* 4. (0+1)*00 (0+1)* 5. 1*(01*01*)*		CO1	L2
	c. Execute Left-Factoring on following grammars: 1. S → 0S1|01 2. S → iEtS|iEtSeS|a E → b		CO1	L3
	d. Illustrate Dangling-else ambiguity while designing Compiler.		CO1	L2

Roll No. _____

Pandit Deendayal Energy University

End Semester Examination: November - December 2022

B. Tech. (Computer Science & Engineering)

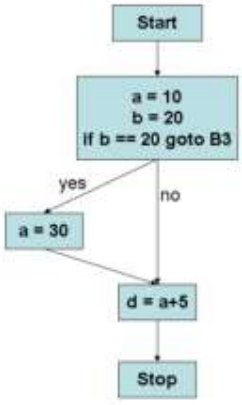
Semester - V

Course Name : System Software And Compiler Design | **Date: 30.11.2022**

Course Code : 20CP302T | **Time: 3 hrs** | **Max. Marks: 100**

Instructions:

1. Do not write anything other than your roll number on question paper.
2. Assume suitable data wherever essential and mention it clearly.
3. Writing appropriate units, nomenclature, and drawing neat sketches/schematics wherever required is an integral part of the answer.

Ques. No.	Description	Marks	CO Mapped	BL
Q.1	**Answer following Questions.**			
	a. What is the difference between a compiler and an interpreter?	01*2=**02**	CO1	L6
	b. Draw the diagram of a compiler front end.			
	c. Differentiate Linker and Loader	03*1=**03**	CO5	
	d. Design annotated parse tree for *3*5+4* using following grammar and using the concept of Syntax Directed Translation (SDT): L->E E->E_1+T E->T T->T_1*F T->F F->(E) F->digit	05*1=**05**	CO1	
	e. How to improve following diagram using constant propagation and folding?  	10*1=**10**	CO1	

Q.2	**Illustrate each and every step of any FOUR (out of following) example.**	10*4=**40**		
	a. Create LL (1) parsing table and parse the string : *int*int* using following grammar 1. P → E 2. E → T E' 3. E' → + T E' 4. E' → ϵ 5. T → F T' 6. T' → * F T' 7. T' → ϵ 8. F → (E) 9. F → int		CO2	L6
	b. Create Shift-Reduce Parsing table for the following grammar and parse the string *id(id+id)* 1. P → E 2. E → E + T 3. E → T 4. T → id (E) 5. T → id			
	c. Crate LR(0) collection of items for following grammar 1. P → E 2. E → E + T 3. E → T 4. T → id (E) 5. T → id			
	d. Create SLR parsing table and trace the string *id(id+id)* using following grammar. 1. P → E 2. E → E + T 3. E → T 4. T → id (E) 5. T → id			
	e. Create LR(1) Collection of items for the following grammar 1. P → E 2. E → E + T 3. E → T 4. T → id (E) 5. T → id			

Q.3	**Answer following Questions (Any FIVE).**	10*4=**40**		
	a. Illustrate these methods using relevant examples in peephole optimization: (1) Redundant Load and Store (2) Strength Reduction		CO4	L6
	b. Design 3AC, Quadruples, Triples, Syntax Tree and DAG for this input: *a + b * c – d / (b * c)*		CO3	
	c. Consider the following expression and construct a DAG for it- *(((++a + a) + (a + a)) + ((a++ + a) + (a + a)))*		CO3	
	d. What do you means by S-attributed SDT and L-attributed SDT? Explain using suitable examples.		CO2	
	e. What do you mean by one pass and two pass Assembler? Explain it in detail.		CO6	

Roll No. _____

Pandit Deendayal Energy University

Re - Exam May -2023

B. Tech. (Computer Science & Engineering)

Semester - V

Course Name : System Software And Compiler Design | **Date: 16.05.2023**

Course Code : 20CP302T | **Time: 3 hrs** | **Max. Marks: 100**

Instructions:

1. Do not write anything other than your roll number on question paper.
2. Assume suitable data wherever essential and mention it clearly.
3. Writing appropriate units, nomenclature, and drawing neat sketches/schematics wherever required is an integral part of the answer.

Ques. No.	Description	Marks	CO Mapped	BL
Q.1	**Answer following Questions.**			
	a. Present a Parse Tree for *Jane Sees Spot Run* using following grammar: *sentence* → *subject predicate* *subject* → *noun* *predicate* → *verb object* *object* → *noun opt_participle* *opt_participle* → *participle* | ε *noun* → **SPOT | JANE | DICK** *participle* → **RUN** *verb* → **SEES**	02*1=**02**	CO2	L3
	b. Present a structure of a Typical Four-Pass Compiler where pass 1 is preprocessor, pass 3 is optimization and pass 4 is back end.	03*1=**03**	CO1	
	c. Classify the Precedence from 1-highest to 5-lowest for following five operators: ( ) [ ] * + ? cc |	05*1=**05**	CO2	L4
	d. Design a Parse Tree for *1+2* (3+4) +5* using following grammar: 1. *stmt* → *expr* ; 2. | *expr* ; *stmt* 3. *expr* → *expr* + *term* 4. | *term* 5. *term* → *term* * *factor* 6. | *factor* 7. *factor* → **number** 8. | (*expr*)	10*1=**10**		L6
Q.2	**Answer following Questions (Any FOUR).**	10*4=**40**		
	a. Present the difference between Linker and Loader.		CO5	L3

	b. Calculate first and follow set for following grammar: 1: *statement* → **OPEN_CURLY** *expression* **CLOSE_CURLY** 2: | *expression* **SEMICOLON** 3: *expression* → **OPEN_PAREN** *expression* **CLOSE_PAREN** 4: | *term* **MINUS** *expression* 5: | ε 6: *term* → **NUMBER** 7: | **IDENTIFIER**		CO2	
	c. Illustrate the tasks of an assembler.		CO6	
	d. How to Eliminating Left Recursion? Explain it in detail with proper example.		CO2	L2
	e. Design a Bottom-Up Parse of *1*(2+3)* using following grammar: 0. $s \rightarrow e$ 1. $e \rightarrow e + t$ 2. | t 3. $t \rightarrow t * f$ 4. | f 5. $f \rightarrow (e)$ 6. | **NUM**			L6
Q.3	**Answer following Questions (Any FOUR).**	10*4=**40**		
	a. Generate LR (0) State-Transition Table (with Shift-Reduce Table and Goto Table) for following Grammar: 0. $s \rightarrow e$ 1. $e \rightarrow e + t$ 2. | t 3. $t \rightarrow t * f$ 4. | f 5. $f \rightarrow (e)$ 6. | **NUM**		CO2	L6
	b. Generate LR (l) State Machine for Grammar of Q-3 (a).			
	c. Explain the characteristics, a symbol-table manager must have.		CO3	L2
	d. Explain Linear (Peephole) Optimizations with proper example.		CO4	

	e. The multiplication by small numbers can be replaced by multiple additions: t0*=3 can be replaced with
	t1=t0;
	t0 += t1;
	t0 += t1;
	Give the name of above Reduction and explain that in detail.			

Roll No. _____

Pandit Deendayal Petroleum University

Take Home Assignment
B. Tech. (Computer Science and Engineering)
Semester - VI

Date:12/06/2020

Course Name : System Software & Compiler Design
Course Code : 18CP312T **Max. Marks:50**

Instructions:

1. Submit hand written assignment on foolscape A4 size pages.
2. Write your name, roll no., subject name and code at top of the assignment.
3. Assume suitable data wherever essential and mention it clearly.
4. Writing appropriate units, nomenclature, and drawing neat sketches/schematics wherever required is an integral part of the answer.
5. Submit assignment online through TCSiON only for regular students and on LMS (Moodle) for backlog students only.

Part-A

ANSWER ALL QUESTIONS (5 x 4 Marks = 20 Marks)

Question No.	Description	Marks	Course Outcome (CO)
Que-1	Considering the following expression –(a+b)*(c-d)+(a+b*c) design the intermediate code and represent it into different forms of 3 address code	04	CO6

Intermediate Code	Quadruple	Triple	Indirect Triple

Que-2	Design the SDT to convert the following binary number to decimal no 110011.1001 also calculate the yield of it.	04	CO4
Que-3	For the following program, derive the machine code	04	CO6

```
        START 101
        READ N

        MOVER BREG, ONE
        MOVEM BREG, TERM

AGAIN   MULT BREG, TERM
        MOVER CREG, TERM
        ADD CREG, ONE
        MOVEM CREG, TERM
```

```
                COMP CREG, N
                BC LE, AGAIN
                MOVEM BREG, RESULT
                PRINT RESULT
                STOP
N               DS 1
RESULT          DS 1
ONE             DC '1'
TERM            DS 1
                END
```

Que-4 Design the regular expression to validate an IP address. 04 CO6

Input: str = "000.12.12.034", Output : True
Input: str = "000.12.234.23.23" Output: False
Input: str = "121.234.12.12" Output: True
Input: str = "I.Am.not.an.ip" Output: False

Que-5 Execute back-patching on the following expression 04 CO6
(x<10) || (x > 200) && (x != y) || (y == 0)

Part-B

ANSWER ALL QUESTIONS (3 x 10 Marks = 30 Marks)

Que-1 Build the program flow graph for the following program: 10 CO5

```
[10]
z=5;
w=z;
for i =1 to 100 do
        x=a*b;
        y=c+d;
        if y<0 then
                a=25;
                f=c+d;

        else
                g=w;
                h=a*b+f;
                d=z+10;
end
g=c+d;

print g, h, d, x, y;
```

Apply the following transformations to optimize the program:

a. Common subexpression elimination
b. Dead code elimination
c. Constant propagation
d. Frequency reduction

Que-2 How do you automatically fuse conditionals into the loop index, optimize the following codes? 10 CO5

for(i=0;i<n;i++){
if(i > 3 && i < 8)
{ p(i); }
}
	for(i=0;i<n;i++){
	if(isodd(n))
	{p(i);}
	}
		for(i=0;;i++){
		if((i>>2)%2 == 0)
		{ p(i); }
		}

Que-3

Un-Optimized code	Optimized code (write the technique and optimized code)
S1 = 4 x i S2 = a[S1] S3 = 4 x j S4 = 4 x i // S5 = n S6 = b[S4] + S5	

for (int j = 0 ; j < n ; j ++)
{
x = y + z ;
a[j] = 6 x j;
}	

i = 0 ;
if (i == 1)
{
a = x + 5 ;
}	

B = A x 2	

10 CO5

Roll No. _____

Pandit Deendayal Petroleum University

Mid Semester Examination

B. Tech. (Computer Science and Engineering)

Semester - VI

Date: 04/03/2020

Course Name : System Software & Compiler Design **Time: 2 hours**

Course Code : 18CP312T **Max. Marks: 50**

Instructions:

1. Do not write anything other than your roll number on question paper.
2. Assume suitable data wherever essential and mention it clearly.
3. Writing appropriate units, nomenclature, and drawing neat sketches/schematics wherever required is an integral part of the answer.

Q.1	**Answer the Following:**		**[9]**	
1.	List and explain various phases available in compiler with suitable example.		4	CO3
2.	What is the problem with left recursive grammar?		2	CO3
3.	**A→ A+T | T** **A→ aA** **B→ aB** Is the above grammar Non Deterministic Grammar. Justify your answer?		1	CO3
4.	Consider the grammar **S → ABSc|ABc** **BA → AB** **Bb → bb** **Ab → ab** **Aa → aa** Which sentences can be derived by this grammar?		2	CO3
Q.2.	**Answer the Following: (Attempt any Two)**		**[10]**	
1.	Write the predictive parsing algorithm. What measures do you take for erroneous inputs? What do you mean by dangling else program?		5	CO4
2.	For the grammar given below: **S→ AA** **A→ aA | b** Construct LR(0) Parsing table, and parse the input string **aaabb$**		5	CO4
3.	Construct the LL(1) parsing table for the following grammar: **E→ TE'** **E'→ +TE'| ε** **T→ FT'**		5	CO4

T'→ *FT'| ε
F→ id | (E)

Q.3 Answer the Following: **[16]**

1. What are the disadvantages of Operator Precedence Parsing, construct the operator relation table and operator function table considering the following grammar. 8 CO4

P → SR | S
R → bSR | bS
S → WbS | W
W → L*W | L
L → id

2. Check the following grammar for LL(1), LR(0), SLR(1), LALR(1), CLR(1) 8 CO4

S → Aa
| bAc
| Bc
| bBa
A → d
B → d

Also, construct the canonical collection of LR(0) items and canonical collection of LR(1) items

Q.4 Answer the Following: **[15]**

1. **S → aA | *S** 3 CO3
A → +S | (S | ε

Set **{ +, (}** will be in the which one First(A), First (E), Follow (E) or Follow (A)

2. Consider the following grammar 3 CO3

S → aB | aAb
A → bAb | a
B → aB | ε

How many back tracks are required to generate the string **aab** from the above grammar?

3. Identify the number of Shift-Reduce and Reduce-Reduce conflicts in LR(0) items for the following grammar. (Draw collection of LR(0) and derive the outcome) 3 CO3

S→ SS
| a
| ε

4. **S → aSAb | bSBc** 3 CO3
A → +AB | ε
B → *BC | ε
C → aC | d

What is the Follow of S and Follow of B?

5. **P → PαQ | Q** 3 CO4
Q → QβR | R
R → num

If 2α3α4β1α2β1 is evaluated to 18, then what is the correct value for α and β?
The above transition rule is used to evaluate 7α4β2α2β1. What will be the result?

Roll No. _____

Pandit Deendayal Petroleum University

School of Technology

Mid Semester Examination

B. Tech. (Computer Engineering) | **Semester – VI**
Date: 07/03/2019 | **Time: 2.30 pm to 4.30 pm**
Course Name : System Software & Compiler Design | **Course Code : 18CP312T**
Max. Marks: 50

Instructions:

1. Do not write anything other than your roll number on question paper.
2. Assume suitable data wherever necessary and mention your assumptions clearly.
3. Write appropriate units, nomenclature and draw neat sketches/schematics, wherever required.
4. Answer all parts of a question continuously.

Q.1 **Answer the Following** **[9]**

1. Eliminate left recursion from the given grammar. 3

   i.) E→ E+T | T **Solution E→TE', E'→ ε |+TE'**

   ii.) S→ S0S1S | 01 **Solution S→ 01S', S'→| 0S1SS'**

   iii.) S→ A

   A→ Ad | Ac | Af | aB | ac

   B→ bBc | F

   Solution:

   S→ A

   A→aBA' | acA'

   A'→ dA' | cA' | fA' | **ε**

   B→ bBc | F

2. Eliminate Non Determinism from the given Grammar 3

   S→ xSSyS

   | xSxSy

   | xyy

   | y

   Solution: S→ xS'|y

   S' → SS' | yy

   S''→SyS | xSy

3. Give your comments on the statement "Eliminating Non Determinism will eliminate ambiguity or not" 1

   Solution: No ambiguity still remains.

5. Check the ambiguity of the following grammar, prove it using parse tree 2

   1) P→ P+P | PP | P* | a | b | c

Solution: It is Ambiguous grammar take some string like a+bc and draw two parse tree

2) S→ aSbS|bSaS|ε

Solution: It is Ambiguous grammar take some string like abab and draw two parse tree

Q.2. Answer the Following **[10]**

1. Construct the First set and Follow set for the following grammar 5

S→ xBDy {x} {$}

B→cC {c} {y, z, w}

C→bC| ε {b, ε} {y, z, w}

D→ EF {z, w, ε} {y}

E→ z | ε {z, ε} {w,y}

F→ w | ε {w, ε} {y}

2. Why we don't want NDG (Non Deterministic Grammar) in TDP (Top Down Parsing)? 1

Solution: As TDP will have to do lot of Back Tracking.

3. Check the following grammar is LL(1) or not? 4

1) S→A|a

A→ a

Solution is Not LL(1) {a} intersection {a}, Also it is ambiguous

2) S→ iEtSS'|a

S'→ eS| ε

E→ b

Solution is Not LL(1) for First(S')={e} intersection Follow(S')={e}, Also it is ambiguous

Q.3 Answer the Following: (Attempt any two) **[16]**

1. Explain Operator Precedence Parser in detail, construct the operator relation table and operator function table considering the grammar 8

E → E + E

| E * E

| id

2. Generate the 3 address Intermediate code for the statement 8

-(x * y) % (x + y) + (x * y + z) also generate Quadruples, Triple and Indirect Triple

3. Check the following grammar for LL(1), LR(0), SLR(1), LALR(1), CLR(1) 8

S → C C

C→ a C | b

Construct the canonical collection of LR(0) items and canonical collection of LR(1) items

Q.4 Answer the Following **[15]**

1. Write the Syntax Directed Translation for calculating the binary value 10101.01 5

2. If A→ LM **{M.i=f(L.i);A.s=f(M.s);1L.i=f(A.i);}** 1
   A→ QR **{R.i=f(A.i); A.s=f(Q.s); Q.i=f(R.i);}**
   a) S-attrib b) L-attrib c) Both d) None
3. Consider SLR(1) and LALR(1) table for a Context Free Grammar identify the correct and incorrect statements. 3
   a) Goto of both tables must be different **(Incorrect)**
   b) Shift entries are identical in both tables **(Correct)**
   c) Reduce entries in tables may be different **(Correct)**
   d) Error entries in table may be different **(Correct)**

4. Identify the number of Shift-Reduce and Reduce-Reduce conflicts in LR(0) items for the following grammar. (Draw collection of LR(0) and derive the outcome) 2
   S→ SS
   | a
   | ε
   Solution 3 SR and 1 RR conflict

5. Differentiate S-Attribute SDT and L-attributed SDT 2
6. What is Back Patching, generate 3 Address code and perform back patching for the following and let the storage starts from address 100. 2
   if (a < b) then t = 1
   else t=0

   Solution
   100 if a < b goto 103
   101 t = 0
   102 go to 104
   103 t = 1
   104

R. **Attendance Record**
Will be collected day by day and updated on TCSIon.

S. Records of the Continuous Assessment (Assignment, Quiz, Laboratory Work, etc.)
As per attachment.

T. Details of Remedial Classes (with evidences)
As per attachment.

U. Details of Expert Lectures / Industrial Visits/Events (Only related to the course)
As per attachment.

V. List of Slow and Advanced Learners, activity planned and executed
As per attachment.

W. Direct Assessment (Result of mid, end and internal assessment components)

X. Indirect Assessment (Exit Survey/Post Test)

Y. Final Attainment of COs and POs and Interpretation

Z. Actions to be taken if COs and POs are not achieved

Will update soon.