SSM INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Approved by AICTE, New Delhi / Affiliated to Anna University, Chennai / Accredited by NAAC)

Dindigul – Palani Highway, Dindigul – 624 002

Department of Computer Science andEngineering

COURSE FILE

SUBJECT CODE & NAME : CS8602- Compiler Design

YEAR / SEMESTER/SECTION : III / VI

BRANCH : CSE

FACULTY IN-CHARGE : N. ANU LAVANYA, AP/CSE

ACADEMIC YEAR : 2022– 2023 (EVEN-SEMESTER)

S.No.	INDEX	REMARKS
1.	Syllabus	✓
2.	Time Table	✓
3.	Academic Calendar	✓
4.	Class Student Name List	✓
5.	Course Plan	✓
6.	Lecture Notes / Course content (Unit-wise)	✓
7.	Previous years university question papers	✓
8.	Advanced / Medium/Slow learners name list with action taken report	✓
9.	Assignments / Case Study/ CBS / Tutorials - relevant documents	✓
10.	Class test QP with Marksheets	–
11.	Mini-projects /Innovation methods in TLP/ Seminars / Expert lecture /Models (if any)	✓
12.	IT question papers with mark sheets / Answer sheets sample (06 nos. for each IT)	✓
13.	Consolidated sheets - CO attainment, CES & Exit survey forms with justification	
14.	Current semester university question paper with answer key and feedback form.	
15.	Continuous Improvement	

Faculty In-Charge **HoD/CSE** **Principal**

CS8602 COMPILER DESIGN L T P C
3 0 2 4

OBJECTIVES:

- To learn the various phases of compiler.
- To learn the various parsing techniques.
- To understand intermediate code generation and run-time environment.
- To learn to implement front-end of the compiler.
- To learn to implement code generator.

UNIT I INTRODUCTION TO COMPILERS 9

Structure of a compiler – Lexical Analysis – Role of Lexical Analyzer – Input Buffering – Specification of Tokens – Recognition of Tokens – Lex – Finite Automata – Regular Expressions to Automata – Minimizing DFA.

UNIT II SYNTAX ANALYSIS 12

Role of Parser – Grammars – Error Handling – Context-free grammars – Writing a grammar – Top Down Parsing - General Strategies Recursive Descent Parser Predictive Parser-LL(1) Parser-Shift Reduce Parser-LR Parser-LR (0)Item Construction of SLR Parsing Table - Introduction to LALR Parser - Error Handling and Recovery in Syntax Analyzer-YACC.

UNIT III INTERMEDIATE CODE GENERATION 8

Syntax Directed Definitions, Evaluation Orders for Syntax Directed Definitions, Intermediate Languages: Syntax Tree, Three Address Code, Types and Declarations, Translation of Expressions, Type Checking.

UNIT IV RUN-TIME ENVIRONMENT AND CODE GENERATION 8

Storage Organization, Stack Allocation Space, Access to Non-local Data on the Stack, Heap Management - Issues in Code Generation - Design of a simple Code Generator.

UNIT V CODE OPTIMIZATION 8

Principal Sources of Optimization – Peep-hole optimization - DAG- Optimization of Basic Blocks- Global Data Flow Analysis - Efficient Data Flow Algorithm.

LIST OF EXPERIMENTS:

1. Develop a lexical analyzer to recognize a few patterns in C. (Ex. identifiers, constants, comments, operators etc.). Create a symbol table, while recognizing identifiers.
2. Implement a Lexical Analyzer using Lex Tool
3. Implement an Arithmetic Calculator using LEX and YACC
4. Generate three address code for a simple program using LEX and YACC.
5. Implement simple code optimization techniques (Constant folding, Strength reduction and Algebraic transformation)
6. Implement back-end of the compiler for which the three address code is given as input and the 8086 assembly language code is produced as output.

PRACTICALS 30 PERIODS
THEORY 45 PERIODS
TOTAL : 75 PERIODS

OUTCOMES:

On Completion of the course, the students should be able to:

- Understand the different phases of compiler.
- Design a lexical analyzer for a sample language.

- Apply different parsing algorithms to develop the parsers for a given grammar.
- Understand syntax-directed translation and run-time environment.
- Learn to implement code optimization techniques and a simple code generator.
- Design and implement a scanner and a parser using LEX and YACC tools.

TEXT BOOK:

1. Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman, Compilers: Principles,Techniques and Tools‖, Second Edition, Pearson Education, 2009.

REFERENCES

1. Randy Allen, Ken Kennedy, Optimizing Compilers for Modern Architectures: ADependence based Approach, Morgan Kaufmann Publishers, 2002.
2. Steven S. Muchnick, Advanced Compiler Design and Implementation‖, Morgan Kaufmann Publishers - Elsevier Science, India, Indian Reprint 2003.
3. Keith D Cooper and Linda Torczon; Engineering a Compiler‖, Morgan Kaufmann Publishers Elsevier Science, 2004.
4. V. Raghavan, Principles of Compiler Design‖, Tata McGraw Hill Education Publishers, 2010.
5. Allen I. Holub, Compiler Design in C‖, Prentice-Hall Software Series, 1993.

Faculty incharge

G.Lilb

HOD/CSE

SSM INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Approved by AICTE New Delhi, Affiliated to Anna University Chennai, Accredited by NAAC)

Dindigul – Palani Highway, Dindigul – 624 002.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Time Table

AcademicYear 2022-2023 (Even Semester) **Year/Semester: III/VI** **Hall:C-209** **w.e.f.:02.02.2023**

Day/Hour	1	2	-	3	4	-	5	6	7
	09.00 to 09.45	09.45 to 10.30	10.30 to 10.45	10.45 to 11.30	11.30 to 12.15	12.15 to 01.00	01.00 to 01.45	01.45 to 02.30	02.30 to 03.15
Monday	CS8661/CS8662		BREAK	CS8661/CS8662		LUNCH BREAK	CS8601	CS8602	CS8691
Tuesday	CS8602			CS8651	CS8603		CS8651	HS8581	
Wednesday	CS8651	CS8602		CS8601	CS8603		CS8602	CS8651	Ment.
Thursday	CS8661/CS8662			CS8661/CS8662			CS8691	CS8603	CS8601
Friday	CS8611			CS8691	CS8601		CS8651	CS8691	CS8603
Saturday	CS8601	CS8691		CS8651	Lib.		CS8602	Pl.T.	

Details of Subjects and Faculty

Sub. Code	Subject Name	Faculty Name/Design/Dept.	No. of hrs/ week	Sub. Code	Subject Name	Faculty Name/Design/Dept.	No. of hrs/ week
CS8651	Internet Programming	Mrs.K.Sureka, AP/CSE	6	CS8662	Mobile Application Development Laboratory	Mr.G.Murugan, AP/CSE	4
CS8691	Artificial Intelligence	Ms.N.Padma Priya, AP/CSE	5	CS8611	Mini Project	Ms. M. Moohambikai, AP/CSE	2
CS8601	Mobile Computing	Mr.G.Murugan, AP/CSE	5	HS8581	Professional Communication	Ms.P. Kothai Natchiar	2
CS8602	Compiler Design	Ms.N.Anu Lavanya, AP/CSE	6	Ment.	Mentoring	Dr.G.Prabu, ASP/CSE, Ms.K.Sureka, AP/CSE, Ms.N.J.Divya, AP/CSE	1
CS8603	Distributed Systems	Ms.N.J.Divya, AP/CSE	4	Lib.	Library		1
CS8661	Internet Programming Laboratory	Ms.K.Sureka, AP/CSE	4	Pl.T	Placement Training	Mr.G.Murugan, AP/CSE	2
Cumulative Hrs			30	Cumulative Hrs			12
Total Hrs/Week							42

Time Table I/C
(M.Moohambikai AP/CSE)

Class In-charge
(Ms.K.Sureka, AP/CSE)

HoD/CSE
(Dr.C.Sujatha, Prof. /CSE)

Principal
(Dr. D. Senthil Kumaran)

SSM INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Approved by AICTE New Delhi, Affiliated to Anna University Chennai, Accredited by NAAC)
Dindigul – Palani Highway, Dindigul – 624 002.
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Time Table

AcademicYear 2022-2023 (Even Semester) Year/Semester: III/VI Hall:C-209 w.e.f.:20.03.2023

Day/Hour	1	2	-	3	4	-	5	6	7
	09.00 to 09.45	09.45 to 10.30	10.30 to 10.45	10.45 to 11.30	11.30 to 12.15	12.15 to 01.00	01.00 to 01.45	01.45 to 02.30	02.30 to 03.15
Monday	CS8661/CS8662		BREAK	CS8661/CS8662		LUNCH BREAK	CS8603	CS8601	CS8691
Tuesday	CS8602			CS8651	CS8603		CS601	HS8581	
Wednesday	CS8661/CS8662			CS8661/CS8662			CS8602	CS8651	CS8602
Thursday	Naan Mudhalvan			Naan Mudhalvan			Naan Mudhalvan		
Friday	CS8611			CS8691	CS8601		CS8651	CS8691	CS8603
Saturday	CS8601	CS8691		CS8651	Lib./ Ment.		CS8602	CS8603	Pl.T.

Details of Subjects and Faculty

Sub. Code	Subject Name	Faculty Name/Design/Dept.	No. of hrs/ week
CS8651	Internet Programming	Mrs.K.Sureka, AP/CSE	4
CS8691	Artificial Intelligence	Ms.N.Padma Priya. AP/CSE	4
CS8601	Mobile Computing	Mr.G.Murugan, AP/CSE	4
CS8602	Compiler Design	Ms.N.Anu Lavanya, AP/CSE	5
CS8603	Distributed Systems	Ms.N.J.Divya, AP/CSE	4
CS8661	Internet Programming Laboratory	Ms.K.Sureka, AP/CSE	4
CS8662	Mobile Application Development Laboratory	Mr.G.Murugan, AP/CSE	4
Cumulative Hrs			29

Sub. Code	Subject Name	Faculty Name/Design/Dept.	No. of hrs/ week
CS8611	Mini Project	Ms. M. Moohambikai, AP/CSE Mrs.S.Suganya, AP/CSE	2
HS8581	Professional Communication	Ms.P. Kothai Natchiar	2
Ment./ Lib.	Mentoring / Library	Dr.G.Prabu, ASP/CSE, Ms.K.Sureka, AP/CSE, Ms.N.J.Divya, AP/CSE	1
Pl.T	Placement Training	Mr.G.Murugan, AP/CSE	1
Naan Mudhalvan	Naan Mudhalvan	Dr.G.Prabu, ASP/CSE, Mrs.Abirami, AP/ECE	7
Cumulative Hrs			13
Total Hrs/Week			42

Time Table I/C
(MrsM.Moohambikai AP/CSE)

Class In-charge
(Ms.K.Sureka, AP/CSE)

HoD/CSE
(Dr.C.Sujatha, Prof. /CSE)

Principal
(Dr. D. Senthil Kumaran)

SSM INSTITUTE OF ENGINEERING AND TECHNOLOGY

DINDIGUL-PALANI HIGHWAY, DINDIGUL-624 002

CALENDAR FOR THE ACADEMIC YEAR 2022-2023 EVEN SEM (UG - IV, VI, VIII SEMESTERS)

January 2023			February 2023			March 2023			April 2023			May 2023			June 2023		
DATE	DAY	SCHEDULE	DATE	DAY	SCHEDULE	DATE	DAY	SCHEDULE	DATE	DAY	SCHEDULE	DATE	DAY	SCHEDULE	DATE	DAY	SCHEDULE
1	SUN	New Year's Day	1	WED	Dept. Academic Advisory Committee Meeting	1	WED		1	SAT	DSM-3	1	MON	May Day	1	THU	
2	MON		2	THU		2	THU	Unit II completion	2	SUN		2	TUE		2	FRI	
3	TUE		3	FRI		3	FRI		3	MON		3	WED	Unit V completion	3	SAT	
4	WED		4	SAT		4	SAT		4	TUE	Mahavir Jayanthi	4	THU	IT - 2 (IV Sem)/IT-3 (VI, VIII Sem)	4	SUN	
5	THU		5	SUN	Thaipoosam	5	SUN		5	WED		5	FRI	IT - 2 (IV Sem)/IT-3 (VI, VIII Sem)	5	MON	
6	FRI		6	MON	Commencement of Classes	6	MON	IT-1 (VI & VIII Semesters)	6	THU	CMS for Unit V	6	SAT	IT - 2 (IV Sem)/IT-3 (VI, VIII Sem)	6	TUE	
7	SAT		7	TUE		7	TUE	IT-1 (VI & VIII Semesters)	7	FRI	Good Friday	7	SUN		7	WED	
8	SUN		8	WED		8	WED	IT-1 (VI & VIII Semesters)	8	SAT		8	MON	IT - 2 (IV Sem)/IT-3 (VI, VIII Sem)	8	THU	
9	MON		9	THU		9	THU	IT-1 (VI & VIII Semesters)	9	SUN		9	TUE	IT - 2 (IV Sem)/IT-3 (VI, VIII Sem)	9	FRI	
10	TUE		10	FRI		10	FRI	IT-1 (VI & VIII Semesters)	10	MON		10	WED	IT - 2 (IV Sem)/IT-3 (VI, VIII Sem)	10	SAT	
11	WED		11	SAT		11	SAT	IT-1 (VI & VIII Semesters)	11	TUE	Unit IV completion	11	THU	Sub. of IT-2, 3 Marks / RAW for IT (IV, VI & VIII Sem) Lab Model Exam	11	SUN	
12	THU		12	SUN		12	SUN		12	WED		12	FRI	DSM-5 / Lab Model Exam / Last Working Day	12	MON	
13	FRI		13	MON	CMS for Unit II	13	MON	IT-1 (IV Sem)	13	THU		13	SAT		13	TUE	
14	SAT		14	TUE		14	TUE	IT-1 (IV Sem) / Sub. of IT-1 Marks / RAW for IT-1 (VI & VIII Sem)	14	FRI	Tamil New Years Day	14	SUN		14	WED	
15	SUN	Pongal	15	WED		15	WED	IT-1 (IV Sem)	15	SAT		15	MON	University Practical Examinations	15	THU	
16	MON	Thiruvalluvar Day	16	THU		16	THU	IT-1 (IV Sem)	16	SUN		16	TUE		16	FRI	
17	TUE	Uzhavar Thirunal	17	FRI	Unit I completion / CCM -1	17	FRI	IT-1 (IV Sem)	17	MON	IT-2 (VI & VIII Semesters)	17	WED		17	SAT	
18	WED		18	SAT	DSM-2	18	SAT	IT-1 (IV Sem)	18	TUE	IT-2 (VI & VIII Semesters)	18	THU		18	SUN	
19	THU		19	SUN		19	SUN		19	WED	IT-2 (VI & VIII Semesters)	19	FRI		19	MON	
20	FRI		20	MON		20	MON	Sub. of IT-1 Marks / RAW for IT-1 (IV Sem)	20	THU	IT-2 (VI & VIII Semesters)	20	SAT		20	TUE	
21	SAT		21	TUE		21	TUE	CMS for Unit IV	21	FRI	IT-2 (VI & VIII Semesters)	21	SUN		21	WED	
22	SUN		22	WED		22	WED	Telugu New Year Day	22	SAT	Ramzan	22	MON		22	THU	
23	MON		23	THU		23	THU		23	SUN		23	TUE		23	FRI	
24	TUE		24	FRI		24	FRI	LTP-1 / PM / CCM -2	24	MON	IT-2 (VI Sem)	24	WED		24	SAT	
25	WED		25	SAT		25	SAT		25	TUE	Sub. of IT-2 Marks / RAW for IT-2 (VI & VIII Sem)	25	THU		25	SUN	
26	THU	Republic Day	26	SUN		26	SUN		26	WED		26	FRI	University Theory Examinations	26	MON	
27	FRI		27	MON	CMS for Unit III	27	MON	Unit III completion	27	THU		27	SAT		27	TUE	
28	SAT		28	TUE		28	TUE		28	FRI	LTP-2 / PM / CCM -3	28	SUN		28	WED	
29	SUN					29	WED		29	SAT	DSM-4	29	MON		29	THU	Bakrid
30	MON	CMS for Unit I				30	THU		30	SUN		30	TUE		30	FRI	
31	TUE	DSM-1				31	FRI					31	WED				
Working Days		NA	Working Days		20	Working Days		26	Working Days		20	Working Days		10	Working Days		NA
Cumulative Days		NA	Cumulative Days		20	Cumulative Days		46	Cumulative Days		66	Cumulative Days		76	Cumulative Days		NA

DSM - Department Staff Meeting
IT- Internal Test
CMS- Course Material Submission
CCM - Class Committee Meeting
LTP - Letter to parents
RAW - Remedial Action Work
PM - Parents Meeting
Total No. of working days : 76

	Holiday		Working Day

1) Department Academic Advisory Committee meetings can be conducted between 01.02.2023 and 20.02.2023 as per the convenience of the respective departments.
2) Changes in the Academic calendar (if any) will be intimated to the students through circular

PRINCIPAL

SSM INSTITUTE OF ENGINEERING AND TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Academic Year : 2022-2023 (Even Semester)

Batch:2020-2024 Year: III Semester: VI

Student Namelist

Batch I

S.No	Register Number	Name of the Student
1	922120104001	ABINAYA G
2	922120104002	ABINAYA T
3	922120104003	AKALYA A N
4	922120104004	ANU P
5	922120104005	ASHOK KUMAR M
6	922120104006	ASMETAA G Y
7	922120104007	BALASURYA R
8	922120104008	BHOOMIKA R
9	922120104009	DINESH RAJA E
10	922120104010	EVANS ABRAHAM J
11	922120104011	HAREESWARAN S
12	922120104012	HARINI S
13	922120104013	JEEVA J
14	922120104014	JEYASHREE S
15	922120104015	JOHANS PRAVEEN S
16	922120104016	JOSEPHINE JESILA M
17	922120104018	KARUNYA M D
18	922120104019	KARUPPAIYA M
19	922120104022	LOGATHARANI S
20	922120104023	LOGESHWARI P
21	922120104024	LOKESH G
22	922120104025	MADHESH KUMAR D
23	922120104026	MOHAMED ARSATH M
24	922120104027	MOHAMED FAZIL J
25	922120104028	MOHAMED HADHI S
26	922120104029	MUGILAN M
27	922120104030	PARTHI PRASATH N
28	922120104031	PIRUTHVI RAMANA V
29	922120104032	POOJA M

Batch II

S.No	Register Number	Name of the Student
30	922120104033	PRADEEP V
31	922120104034	PRAKASH S
32	922120104035	PREETHIGA M
33	922120104036	PRETHEEBA U
34	922120104037	REENA M
35	922120104038	SABARIKRISHNAN R
36	922120104039	SAKTHI VIGNESHWARAN B
37	922120104040	SANJAY PANDI M
38	922120104041	SANJEEV SARAVANAN S
39	922120104042	SANTHIYADHARSHINI S
40	922120104043	SANTHOSH R
41	922120104044	SARAN PANDIAN S
42	922120104045	SATHEESH KUMAR K
43	922120104046	SHALINI J
44	922120104047	SHARMILA S
45	922120104049	SINDHUJA INFANT A
46	922120104050	SIVA SHANTHANA BHARATH M
47	922120104051	SIVASUNDAR V
48	922120104052	SOUNDHARYA DEVI M
49	922120104053	SRIDHARAN S
50	922120104054	SRIRAM J
51	922120104055	SRIRAM S
52	922120104056	SRIRAM PRASATH L
53	922120104057	SUBBIRAMANI R
54	922120104058	SUBHA S
55	922120104059	TAMIL ARASAN K
56	922120104060	VARSHINI U
57	922120104061	VINOTH KUMAR A
58	922120104301	SRIRAM VM

2/2/23

Class In-Charge

Mrs.K.sureka,AP/CSE

2/2/23

HOD/CSE

Dr.C.Sujatha

Principal

Dr.D.Senthil Kumaran

SSM INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Approved by AICTE, New Delhi / Affiliated to Anna University, Chennai / Accredited by NAAC)
Dindigul – Palani Highway, Dindigul – 624 002

Department of Computer Science and Engineering

List of Advanced and Slow Learners

Academic Year : 2022-2023(EVEN)

Subject Code & Name:CS8602- Compiler Design

Class & Semester & Dept: III & VI Sem & CSE

Faculty In-charge: Ms N. Anu Lavanya , AP/CSE

Advanced Learners

Sl. No.	Register Number	Student Name
1.	922120104008	BHOOMIKA R
2.	922120104024	LOKESH G
3.	922120104032	POOJA M
4.	922120104033	PRADEEP V
5.	922120104047	SHARMILA S

Slow Learners

Sl. No.	Register Number	Student Name
1.	922120104001	ABINAYA G
2.	922120104009	DINESH RAJA E
3.	922120104011	HAREESWARAN S
4.	922120104061	VINOTH KUMAR A
5.	922120104301	SRIRAM V M

Action Taken for Advance Learners

- Motivated students to take seminar.
- Advanced learners were asked to refer NPTEL/ other MOOC course contents on Compiler Design to have a better insight about the subject.

Action Taken for Slow Learners

- Posted handwritten notes in GCR for easy learning.
- Given important question unit wise.
- Have taught the important concepts again for slow learners.

Faculty In-Charge

HoD/CSE

SSM INSTITUTE OF ENGINEERING AND TECHNOLOGY

Dindigul- Palani Highway, Dindigul – 624 002.

(Approved by AICTE, Affiliated to Anna University, Accredited by NAAC)

Department of Computer Science and Engineering

Subject Code: CS8602 –Compiler Design **Year/Sem: III/VI**

Date: 18.02.23

ASSIGNMENT (CO1)

1. **Specify how the following statement will be translated into every phase**
   - i. **x=y+z*60**
   - ii. **a=b*c-d**
2. Describe the Input buffering techniques in detail.
3. Prove that the following two regular expressions are equivalent by showing that the minimum state DFA's are same.
   - i. (a/b)*
   - ii. (a*/b*)*
4. Construct DFA for the following RE using syntax tree with firstpos, lastpos, and followpos function. (a|b)*a.
5. Write the LEX program for the following
   - i. To remove comments from C program.
   - ii. To add line number to a given file
   - iii. To check valid email
   - iv. To check whether the input is digit or not.

Date of Submission: 24.02.23

Faculty in-charge

G. Hi/B

HOD/CSE

Compiler Design

Assignment - 1

Abinaya.T
922120104002

1) Specify how the following statement will be translated into every phase

i) $x = y + z * 60$

$x = y + z * 60$

↓

LA

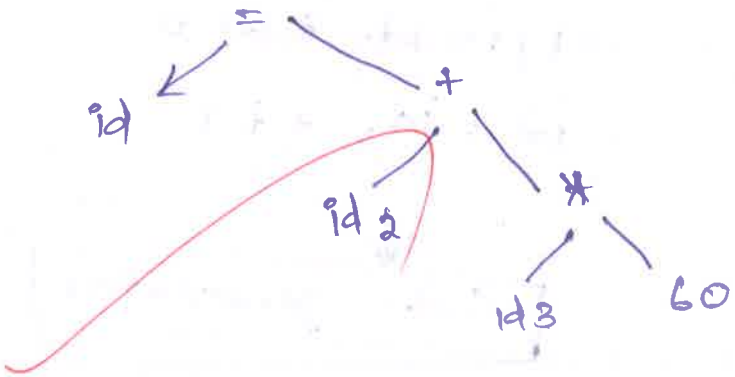
↓

id 1 = id 2 + id 3 * 60

↓

Syntax analyzer

↓



Semantic analyzer

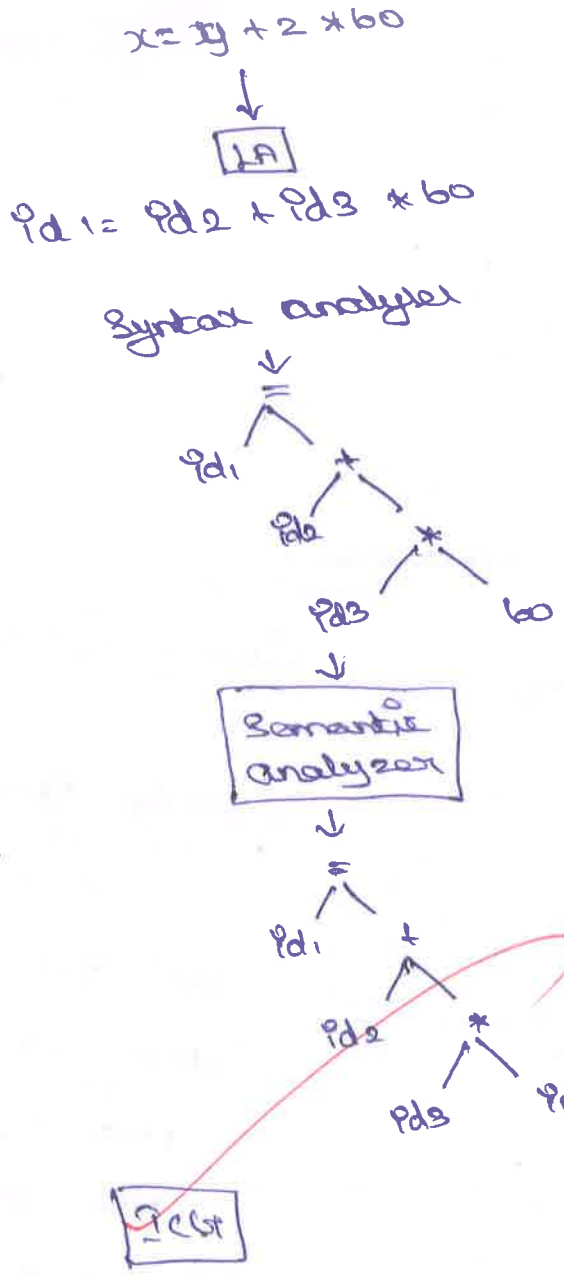
↓

9221201040 53
S. Sridharan
3rd - CSE

Compiler Design

Assignment -1

1) x = y + z * 60



23

Compiler Design

Assignment - 1

20

D. madhesh kumar

9221 2010 4025

CSE

SSM INSTITUTE OF ENGINEERING AND TECHNOLOGY

Dindigul- Palani Highway, Dindigul – 624 002.

(Approved by AICTE, Affiliated to Anna University, Accredited by NAAC)

Department of Computer Science and Engineering

Subject Code: CS8602 –Compiler Design **Year/Sem: III/VI**

Date: 01.03.2023

ASSIGNMENT (CO2)

1. Construct the SLR(1) parsing table for,

   E->E+T E->T

   T->T*F T->F

   F->(E) F->id
2. Write down the algorithm to eliminate left-recursion and left factoring and apply both of the following grammar.

   E-> E+T|E-T|T

   T->a|b|(E)
3. Construct LR(0) parsing table for the given grammar.

   E->E*B E->B B->1

   E->E+B B->0
4. Construct the LR(1) parsing table for the following grammar.

   S->cc C->ac C->d
5. Consider the following grammar E->T+E|T T->V*T|V V->id

   Write down procedure for non-terminals of the grammar to make a recursive descent parser

Date of Submission: 20.03.2023

Faculty in-charge

6.2.18

HOD/CSE

CS8601 - Compiler Design.

Assignment - 2

922120104033

V. Pradeep.

1. Construct the SLR(1) parsing table for,

1) $E \to E+T$ 2) $E \to T$

3) $T \to T*F$ 4) $T \to F$

5) $F \to (E)$ 6) $F \to id$.

The set of items generated by this method are also called SLR(0) items.

I_0 :

$E' \to .E$

$E \to .E+T$

$E \to .T$

$T \to .T*F$

$T \to .F$

$F \to .(E)$

$F \to .id$.

goto(I_0, E)

I_1 : $E' \to E.$

$E \to E.+T$

goto(I_0, T)

I_2 : $E \to T.$

$T \to T.*F$

goto(I_0, F)

I_3 : $T \to F.$

goto(I_0, ()

I_4 : $T \to (.E)$

$E \to .E+T$

$E \to .T$

$T \to .T*F$

goto(I_0, id)

I_5 : $F \to id.$

goto(I_1, +)

I_6 : $E \to E+.T$

$T \to .T*F$

$T \to .F$

$F \to .(E)$

$F \to .id.$

goto(I_2, *)

I_7 : $T \to T*.F$

$F \to .(E)$

$F \to .id.$

goto(I_4, E)

I_8 : $F \to (E.)$

$E \to E.+T$

goto(I_6, T)

I_9 : $E \to E+T.$

$T \to T.*F$

goto(I_7, F)

I_{10} : $T \to T*F.$

goto(I_8,))

I_{11} : $F \to (E).$

25

CS8601 - Compiler Design.

Assignment - 2

922120104033

V. Pradeep.

1. Construct the SLR(1) parsing table for,

1) $E \to E+T$ 2) $E \to T$

3) $T \to T*F$ 4) $T \to F$

5) $F \to (E)$ 6) $F \to id$.

The set of items generated by this method are also called SLR(0) items.

I_0 :

$E' \to .E$

$E \to .E + T$

$E \to .T$

$T \to .T*F$

$T \to .F$

$F \to .(E)$

$F \to .id$.

goto(I_0, E)

I_1 : $E' \to E.$

$E \to E. + T$

goto(I_0, T)

I_2 : $E \to T.$

$T \to T. * F$

goto(I_0, F)

I_3 : $T \to F.$

goto(I_0, ()

I_4 : $T \to (.E)$

$E \to .E+T$

$E \to .T$

$T \to .T*F$

goto(I_0, id)

I_5 : $F \to id.$

goto(I_1, +)

I_6 : $E \to E+.T$

$T \to .T*F$

$T \to .F$

$F \to .(E)$

$F \to .id.$

goto(I_2, *)

I_7 : $T \to T*.F$

$F \to .(E)$

$F \to .id.$

goto(I_4, E)

I_8 : $F \to (E.)$

$E \to E. + T$

goto(I_6, T)

I_9 : $E \to E+T.$

$T \to T. * F$

goto(I_7, F)

I_{10} : $T \to T*F.$

goto(I_8,))

I_{11} : $F \to (E).$

M. Lokesh
9221201040 24

CD Assignment

1). construct the SLR(1) parsing table for

1) $E \to E+T$ 2) $E \to T$

3) $T \to T*F$ 4) $T \to F$

5) $F \to (E)$ 6) $F \to id$.

24

I_0:

$E' \to .E$

$E \to .E+T$

$E \to .T$

$T \to .T*F$

$T \to .F$

$F \to .(E)$

$F \to .id$

goto(I_0, E)

I_1 : $E' \to E.$

$E \to E.+T$

goto(I_0, T)

I_2 : $E \to T.$

$T \to T.*F$

goto(I_0, F)

I_3 : $T \to F.$

goto(I_0, id)

I_5 : $F \to id.$

goto(I_1, +)

I_6 : $E \to E+.T$

$T \to .T*F$

$T \to .F$

$F \to .(E)$

$F \to .id$

goto (I_2, *)

I_7 : $T \to T*.F$

$F \to .(E)$

$F \to .id$

goto (I_4, E)

I_8 : $F \to (E.)$

$E \to E.+T$

goto(I_6, T)

I_9 : $E \to E+T.$

$T \to T.*F$

goto(I_7, F)

I_{10} : $T \to T*F.$

goto (I_8,))

I_{11} : $F \to (E).$

SSM INSTITUTE OF ENGINEERING AND TECHNOLOGY

Dindigul- Palani Highway, Dindigul – 624 002.

(Approved by AICTE, Affiliated to Anna University, Accredited by NAAC)

Department of Computer Science and Engineering

Subject Code: CS8602 –Compiler Design **Year/Sem: III/VI**

Date: 25.03.2023

ASSIGNMENT (CO3)

1. Construct a decorated parser tree according to syntax directed definition for the following input statement (4+7.5)*/2
2. For the following given grammar construct the syntax directed definition and generated the code fragment using S- attributed definition S->EN E->E+T E-> E-T E->T T-> T*F T->T/F T->F F->(E) F->DIGIT N->: also evaluated the input string 2*3+4with parser state using LR parsing method or syntax directed definition for desktop calculator.
3. Obtain the translation schema obtaining the tree address code form grammar.
   S->ID: =E E->E1+E2 E->E1*E2 E->-E1 E->(E1) E->id
4. Boolean expression
5. Using Back patching generate an intermediate code for the following expression
   A<B OR C<D AND P<Q

Date of Submission: 05.04.2023

Faculty In-charge

HOD/CSE

CS8602 - COMPILER DESIGN

ASSIGNMENT - 3

M. SOUNDHARYA DEVI
922120104052
CSE-III

(1) Construct a decorated Parse tree according to the Syntax directed definition for the following input statement: (4 + 7.5 * 3) / 2

Solution: we will write the production rules and semantic actions as follows:

25

Production rule	Semantic actions
$S \rightarrow E$	print (E.val)
$E \rightarrow E+T$	E.val := E_1.val + T.val
$B \rightarrow T$	E.val := T.val
$I \rightarrow T * P$	T.val := T_1.val * P.val
$T \rightarrow P$	T.val = P.val
$P \rightarrow P / F$	P.val := P_1.val / F.val
$P \rightarrow F$	P.val := F.val
$F \rightarrow (E)$	F.val := E.val
$F \rightarrow$ digit	F.val := digit.lexval

Akalya.A.N

9221201040003

Assignment-3

Compiler Design

1. Construct a decorated parse tree according to syntax directed definitions for the following input statement : (4+7.5) * 12

We will write the production rules and semantic actions as follows.

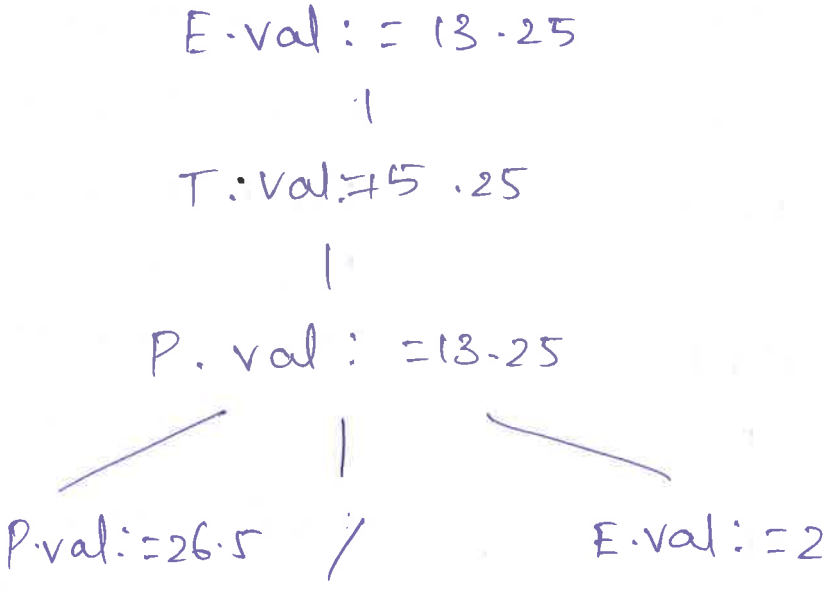
Production rule	Semantic actions
S → E	print (E.val)
E → E + T	E.val := E1.val + T.val
E → T	E.val := T.val
B → T	
T → T * P	T.val := T1.val * P.val
T → P	T.val = P.val
P → P/F.	P.val := P.val, F.val
P → F	P.val := F.val
R → (E)	F.val := E.val
F → digit	F.val := digit.lexval

Asmeeta. G. Y
9221201040006.

Complier design
Assignment-3.

1. We will write the production rules and semantic actions as follows.

Production rules	semantic actions
$S \rightarrow E$	Print (E.val)
$E \rightarrow E+T$	E.val := E_1.val + T.val
$E \rightarrow T$	E.val := T.val
$T \rightarrow T*P$	T.val := T_1.val * P.val
$T \rightarrow P$	T.val = P.val
$P \rightarrow P/F$	P.val := P.val, F.val
$P \rightarrow F$	P.val := F.val
$F \rightarrow (E)$	F.val := E.val.



SSM INSTITUTE OF ENGINEERING AND TECHNOLOGY
Dindigul- Palani Highway, Dindigul – 624 002.
(Approved by AICTE, Affiliated to Anna University, Accredited by NAAC)

Department of Computer Science and Engineering

Subject Code: CS8602 –Compiler Design **Year/Sem: III/VI**

Date: 08.04.2023

ASSIGNMENT (CO4)

1. Storage allocation strategies
2. Generate code for the following statement for target machine.
   N=n+1 N=a+b+c N=a/b-c-d*(e+f)
3. For the following expression obtain optimal code using only two register (a+b)-(c+d+e)
4. Generate optimal code for following assignment x=a+b+c x=(a*-b)+(c-d+e) x=(a/b-c)/d x=a+(b+c|d*e)|(f*g-h*i)
5. By taking the example of factorial program explain how activation record will look like for every receive call of factorial (3).

Date of Submission: 15.04.2023

Faculty In-charge

HOD/CSE

Compiler Design.

R. Santhosh
922120104043

Assignment - 4

25

1) Storage Allocation strategies:-

1. Static allocation - The static allocation is for all the data or the data objects at compile time.

2. Stack allocation - In the stack allocation a stack is used manage the run time storage.

3. Heap Allocation:- In heap allocation the heap is used to manage the dynamic memory allocation.

Static Allocation:-

* The size of data objects is known at compile time. The names of these objects are bound to storage at compile time only and such an allocation of data objects is done by static allocation.

* The binding of name with the amount of storage allocated do not change at run time.

* At compiler time compiler can fill the address at which the target code can find the data it operates on.

* FORTRAN uses the static allocation strategy.

COMPILER DESIGN

20

ASSIGNMENT - 4

9221201040 13

J. Jeeva.

Jeeva - 1

S. SRIRAM.
922120104055

COMPILER DESIGN
ASSIGNMENT

25

1. Source of Optimization.

The Optimization can be done locally or globally. If the transformation is applied on the same basic clock than that kind of transformation is done locally Otherwise transformation is done globally. General the local tranformation are done

Compile Time Evaluation.

Compile time evaluation means shifting of Compulation from run time to Compilation time. There are two methods used to Obtain the Compile time evaluation.

* Folding.

In tail folding technique, the Computation of Constant is done at Compile time instead of execution time.

eg: length = (22/7) * d.

* Constant Propagation.

The Value of Variable is replaced and Computation of an expression is done at the Compitation time.

eg: Pi = 3.14; r = 5;

Area = Pi * r * r.

The Value of Pi replaced by 3.14 & r by 5 then the equ of 3.14 * r * r is done during Complitation

Complier Design

M. Reena.
922120104037.

Principal sources of optimization.

The optimization can be done locally or globally. If the transformation is applied on the same basic block then that kind of transformation locally otherwise is done globally.

22

1. Compile time Evaluation.

Compile time evaluation means shifting of computations from run time to compilation time. There are two methods used to obtain the compile time evaluation.

1. Floding:-

In the floding technique the computation of constant is done at compile time instead of execution time

example :- length = (2217)* d.

2) Constant propagation:

In this technique the value of variable is replaced and computation of an expression is done at the compilation time.

ex:- pi = 3.14;

r = 5;

Area = pi * r * r

Compiler Design
Assignment

Satheeshkumar.K
922120104045

1) Principal Sources of Optimization:

The optimization can be done locally or globally. If the transformation is applied on the same basic block then that kind of transformation is done locally otherwise transformation is done globally. General the local transformations are done first.

20

=> Compile Time Evaluation:

compile time evaluation means shifting of computations from run time to compilation time. There are two methods used to obtain the compile time evaluation.

* folding:

In this folding technique, the computation of constant is done at compile time instead of execution time.

eg: length = (22/7) * d

* Constant Propagation:

The value of variable is replaced and computation of an expression is done at the compilation time.

eg: Pi = 3.14;
r = 5;
Area = Pi * r * r

The value of Pi replaced by 3.14 and r by 5 then the eqn of 3.14 * r * r is done during compilation

SSM INSTITUTE OF ENGINEERING AND TECHNOLOGY

Dindigul- Palani Highway, Dindigul – 624 002.

Department of Computer Science and Engineering

(Assignment Mark statement)

Dept: CSE Subject Code/Name: CS8602-Compiler Design Year/Sem: III/VI

S.No	Reg.No	Name	ASS-1 (Out of 25)	ASS-2 (Out of 25)	ASS-3 (Out of 25)	ASS-4 (Out of 25)	ASS-5 (Out of 25)
1	922120104001	ABINAYA G	23	22	22	24	25
2	922120104002	ABINAYA T	25	25	23	25	25
3	922120104003	AKALYA A N	25	25	22	25	25
4	922120104004	ANU P	25	25	25	25	25
5	922120104005	ASHOK KUMAR M	20	23	22	20	20
6	922120104006	ASMETAA G Y	22	23	19	23	22
7	922120104007	BALASURYA R	20	22	22	23	24
8	922120104008	BHOOMIKA R	25	25	25	25	25
9	922120104009	DINESH RAJA E	24	23	22	24	25
10	922120104010	EVANS ABRAHAM J	25	24	23	24	25
11	922120104011	HAREESWARAN S	25	24	24	24	25
12	922120104012	HARINI S	25	25	25	25	25
13	922120104013	JEEVA J	24	25	24	20	25
14	922120104014	JEYASHREE S	23	24	23	22	23
15	922120104015	JOHANS PRAVEEN S	22	23	25	24	22
16	922120104016	JOSEPHINE JESILA M	24	24	24	24	24
17	922120104018	KARUNYA M D	23	23	22	24	23
18	922120104019	KARUPPAIYA M	25	24	23	24	24
19	922120104022	LOGATHARANI S	25	24	24	24	24
20	922120104023	LOGESHWARI P	24	25	25	25	25

21	922120104024	LOKESH G	25	24	25	25	25
22	922120104025	MADHESH KUMAR D	20	25	25	25	25
23	922120104026	MOHAMED ARSATH M	24	23	22	24	23
24	922120104027	MOHAMED FAZIL J	20	20	20	20	20
25	922120104028	MOHAMED HADHI S	22	23	22	24	23
26	922120104029	MUGILAN M	20	24	23	24	24
27	922120104030	PARTHI PRASATH N	24	24	24	24	24
28	922120104031	PIRUTHVI RAMANA V	23	23	22	24	23
29	922120104032	POOJA M	25	25	25	25	25
30	922120104033	PRADEEP V	25	25	25	25	25
31	922120104034	PRAKASH S	23	23	22	24	23
32	922120104035	PREETHIGA M	22	24	23	24	24
33	922120104036	PRETHEEBA U	24	24	24	24	24
34	922120104037	REENA M	22	23	22	24	23
35	922120104038	SABARIKRISHNAN R	25	21	25	25	25
36	922120104039	SAKTHI VIGNESHWARAN B	25	25	25	25	25
37	922120104040	SANJAY PANDI M	24	23	22	24	23
38	922120104041	SANJEEV SARAVANAN S	20	20	23	22	24
39	922120104042	SANTHIYADHARSHINI S	23	23	22	24	23
40	922120104043	SANTHOSH R	24	24	23	25	24
41	922120104044	SARAN PANDIAN S	23	23	22	24	23
42	922120104045	SATHEESH KUMAR K	24	24	23	24	20
43	922120104046	SHALINI J	25	25	25	25	25
44	922120104047	SHARMILA S	25	25	25	25	25
45	922120104049	SINDHUJA INFANT A	24	23	22	24	23
46	922120104050	SIVA SHANTHANA BHARATH M	20	24	23	24	24

47	922120104051	SIVASUNDAR V	20	23	22	24	23
48	922120104052	SOUNDHARYA DEVI M	22	24	25	24	24
49	922120104053	SRIDHARAN S	23	24	24	24	24
50	922120104054	SRIRAM J	22	23	22	24	23
51	922120104055	SRIRAM S	23	24	20	22	25
52	922120104056	SRIRAM PRASATH L	20	23	24	0	0
53	922120104057	SUBBIRAMANI R	22	23	22	20	22
54	922120104058	SUBHA S	25	25	25	25	25
55	922120104059	TAMIL ARASAN K	24	23	22	23	21
56	922120104060	VARSHINI U	25	25	25	25	25
57	922120104061	VINOTH KUMAR A	10	12	15	20	22
58	922120104301	SRIRAM V M	20	23	22	24	25

FACULTY INCHARGE

HOD/CSE

SSM INSTITUTE OF ENGINEERING AND TECHNOLOGY

DEPARTMENT OF CSE

Circular

12.04.2023

This is to inform all the III Year CSE students, that there will be a Seminar on "**PARSER GENERATOR**" on 15.04.2023 as a part of "**Content beyond syllabus**" session. Henceforth, all the students are advised to attend the class without fail and get benefitted.

Subject: CS8602- Compiler Design

Subject In charge

HoD/CSE

Attendance Report on 15.04.2023

S.NO	REG. NUMBER	Name of the Student	Present/Absent
1.	922120104001	ABINAYA G	Present
2.	922120104002	ABINAYA T	Present
3.	922120104003	AKALYA A N	Present
4.	922120104004	ANU P	Present
5.	922120104005	ASHOK KUMAR M	Present
6.	922120104006	ASMETAA G Y	Present
7.	922120104007	BALASURYA R	Present
8.	922120104008	BHOOMIKA R	Present
9.	922120104009	DINESH RAJA E	Present
10.	922120104010	EVANS ABRAHAM J	Present
11.	922120104011	HAREESWARAN S	Present
12.	922120104012	HARINI S	Present
13.	922120104013	JEEVA J	Present
14.	922120104014	JEYASHREE S	Present
15.	922120104015	JOHANS PRAVEEN S	Present
16.	922120104016	JOSEPHINE JESILA M	Present
17.	922120104018	KARUNYA M D	Present
18.	922120104019	KARUPPAIYA M	Present
19.	922120104022	LOGATHARANI S	Present
20.	922120104023	LOGESHWARI P	Present
21.	922120104024	LOKESH G	Present
22.	922120104025	MADHESH KUMAR D	Present
23.	922120104026	MOHAMED ARSATH M	Present
24.	922120104027	MOHAMED FAZIL J	Present
25.	922120104028	MOHAMED HADHI S	Present

26.	922120104029	MUGILAN M	**Present**
27.	922120104030	PARTHI PRASATH N	**Present**
28.	922120104031	PIRUTHVI RAMANA V	**Present**
29.	922120104032	POOJA M	**Present**
30.	922120104033	PRADEEP V	**Present**
31.	922120104034	PRAKASH S	**Present**
32.	922120104035	PREETHIGA M	**Present**
33.	922120104036	PRETHEEBA U	**Present**
34.	922120104037	REENA M	**Present**
35.	922120104038	SABARIKRISHNAN R	**Present**
36.	922120104039	SAKTHI VIGNESHWARAN B	**Present**
37.	922120104040	SANJAY PANDI M	**Present**
38.	922120104041	SANJEEV SARAVANAN S	**Present**
39.	922120104042	SANTHIYADHARSHINI S	**Present**
40.	922120104043	SANTHOSH R	**Present**
41.	922120104044	SARAN PANDIAN S	**Present**
42.	922120104045	SATHEESH KUMAR K	**Present**
43.	922120104046	SHALINI J	**Present**
44.	922120104047	SHARMILA S	**Present**
45.	922120104049	SINDHUJA INFANT A	**Present**
46.	922120104050	SIVA SHANTHANA BHARATHI M	**Present**
47.	922120104051	SIVASUNDAR V	**Present**
48.	922120104052	SOUNDHARYA DEVI M	**Present**
49.	922120104053	SRIDHARAN S	**Present**
50.	922120104054	SRIRAM J	**Present**
51.	922120104055	SRIRAM S	**Present**
52.	922120104056	SRIRAM PRASATH L	**Present**

53.	922120104057	SUBBIRAMANI R	**Present**
54.	922120104058	SUBHA S	**Present**
55.	922120104059	TAMIL ARASAN K	**Present**
56.	922120104060	VARSHINI U	**Present**
57.	922120104061	VINOTH KUMAR A	**Present**
58.	922120104301	SRIRAM V M	**Present**

FEED BACK ANALYSIS OF PARTICIPANT - SUMMARY

Subject	: CS8602- Compiler Design
Topic	: Parser Generator
Date	: 15.04.2023
Time	: 10.45 a.m – 11.30 a.m.
Resource person (With designation)	: N. Anu Lavanya , AP/CSE, SSMIET
Total no. of students	: 58
No. of students present	: 58
% of students	: 100%

Q1) Did you find the session useful?

a) Very much	35 students
b) To some extent	23 students
c) Not useful	00 students

Q2) Did faculty possess deep knowledge of the subject taught?

a) Yes	40 students
b) To some extent	18 students
c) No	00 students

Q.3) Faculty managed classroom time and pace well

a) Excellent	15 students
b) Very good	15 students
c) Good	15 students
d) Satisfactory	13 students
e) Poor	00 students

REPORT

As a part of Content beyond Syllabus, the III Year CSE students actively participated in the session and got benefitted.

Subject	: CS8602- Compiler Design
Topic	: Parser Generator
Date	: 15.04.2023
Time	: 10.45 a.m – 11.30 a.m.
Resource person (With designation)	: N. Anu Lavanya , AP/CSE, SSMIET
Total no. of students	: 58
No. of students present	: 58
% of students	: 100%
Relevance to POs, PSOs	: PO1, PO2, PO12, PSO1, PSO2.

Content: Attached

Parser Generators

1 The Parser Generator Yacc

2 Using Yacc with Ambiguous Grammars

This section shows how a parser generator can be used to facilitate the construc-tion of the front end of a compiler. We shall use the LALR parser generator **Yacc** as the basis of our discussion, since it implements many of the concepts discussed in the previous two sections and it is widely available. **Yacc** stands for "yet another compiler-compiler," reflecting the popularity of parser generators in the early 1970s when the first version of **Yacc** was created by S. C. Johnson. **Yacc** is available as a command on the UNIX system, and has been used to help implement many production compilers.

1. The Parser Generator Yacc

A translator can be constructed using **Yacc** in the manner illustrated in Fig. 4.57. First, a file, say **translate.y,** containing a **Yacc** specification of the translator is prepared. The UNIX system command

yacc translate.y

transforms the file **translate.y** into a C program called **y.tab.c** using the LALR method outlined in Algorithm 4.63. The program **y.tab.c** is a repre-sentation of an LALR parser written in C, along with other C routines that the user may have prepared. The LALR parsing table is compacted as described in Section 4.7. By compiling **y.tab.c** along with the **ly** library that contains the LR parsing program using the command

cc y.tab.c -ly

we obtain the desired object program **a.** out that performs the translation spec-ified by the original **Yacc** program . [7] If other procedures are needed, they can be compiled or loaded with **y.tab.c,** just as with any C program.

A Yacc source program has three parts:

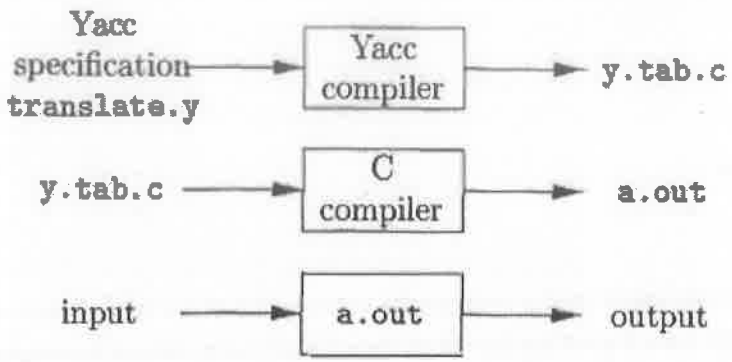


Figure 4.57: Creating an input/output translator with Yacc

declarations
%%
translation rules
%%
supporting C routines

```
%token DIGIT

%{
#include <ctype.h>
%}
%token DIGIT
%%
line    : expr '\n'          { printf("%d\n", $1); }
        ;
expr    : expr '+' term      { $$ = $1 + $3; }
        | term
        ;
term    : term '*' factor    { $$ = $1 * $3; }
        | factor
        ;
factor  : '(' expr ')'       { $$ = $2; }
        | DIGIT
        ;
%%
yylex() {
     int c;
     c = getchar();
     if (isdigit(c)) {
          yylval = c-'0';
          return DIGIT;
     }
     return c;
}
```

Figure 4.58: Yacc specification of a simple desk calculator

$$E \rightarrow E + T \mid T$$

and their associated semantic actions as:

```
expr : expr '+' term    { $$ = $1 + $3; }
     | term
     ;
```

The lexical analyzer in Fig. 4.58 is very crude. It reads input characters one at a time using the C-function get char ( ) . If the character is a digit, the value of the digit is stored in the variable y y l v a l , and the token name DIGIT is returned. Otherwise, the character itself is returned as the token name.

2. Using Yacc with Ambiguous Grammars

Let us now modify the **Yacc** specification so that the resulting desk calculator becomes more useful. First, we shall allow the desk calculator to evaluate a sequence of expressions, one to a line. We shall also allow blank lines between expressions. We do so by changing the first rule to

```
lines : lines expr '\n'    { printf("%g\n", $2); }
      | lines '\n'
      | /* empty */
      ;
```

In **Yacc,** an empty alternative, as the third line is, denotes e.

Second, we shall enlarge the class of expressions to include numbers instead of single digits and to include the arithmetic operators +, —, (both binary and unary), *, and /. The easiest way to specify this class of expressions is to use

the ambiguous grammar

$$E \rightarrow E + E \mid E - E \mid E * E \mid E / E \mid - E \mid \textbf{number}$$

declares DIGIT to be a token. Tokens declared in this section can then be used in the second and third parts of the Yacc specification. If Lex is used to create the lexical analyzer that passes token to the Yacc parser, then these token declarations are also made available to the analyzer generated by Lex, as discussed in Section 3.5.2.

The Translation Rules Part

In the part of the Yacc specification after the first %% pair, we put the translation rules. Each rule consists of a grammar production and the associated semantic action. A set of productions that we have been writing:

$$\langle \text{head} \rangle \rightarrow \langle \text{body} \rangle_1 \mid \langle \text{body} \rangle_2 \mid \cdots \mid \langle \text{body} \rangle_n$$

would be written in Yacc as

$$\begin{array}{lll} \langle \text{head} \rangle & : & \langle \text{body} \rangle_1 \quad \{ \langle \text{semantic action} \rangle_1 \} \\ & | & \langle \text{body} \rangle_2 \quad \{ \langle \text{semantic action} \rangle_2 \} \\ & & \cdots \\ & | & \langle \text{body} \rangle_n \quad \{ \langle \text{semantic action} \rangle_n \} \\ & ; & \end{array}$$

In a Yacc production, unquoted strings of letters and digits not declared to be tokens are taken to be nonterminals. A quoted single character, e.g. ' c', is taken to be the terminal symbol **c**, as well as the integer code for the token represented by that character (i.e., **Lex** would return the character code for ' **c**' to the parser, as an integer). Alternative bodies can be separated by a vertical bar, and a semicolon follows each head with its alternatives and their semantic actions. The first head is taken to be the start symbol.

A **Yacc** semantic action is a sequence of C statements. In a semantic action, the symbol $$ refers to the attribute value associated with the nonterminal of the head, while $i refers to the value associated with the *ith* grammar symbol (terminal or nonterminal) of the body. The semantic action is performed when-ever we reduce by the associated production, so normally the semantic action computes a value for $$ in terms of the $i's. In the **Yacc** specification, we have

Written the two *E*-productions

```
%{
#include <ctype.h>
#include <stdio.h>
#define YYSTYPE double  /* double type for Yacc stack */
%}
%token NUMBER

%left '+' '-'
%left '*' '/'
%right UMINUS
%%

lines : lines expr '\n'   { printf("%g\n", $2); }
      | lines '\n'
      | /* empty */
      ;
expr  : expr '+' expr     { $$ = $1 + $3; }
      | expr '-' expr     { $$ = $1 - $3; }
      | expr '*' expr     { $$ = $1 * $3; }
      | expr '/' expr     { $$ = $1 / $3; }
      | '(' expr ')'      { $$ = $2; }
      | '-' expr  %prec UMINUS { $$ = - $2; }
      | NUMBER
      ;
%%
yylex() {
    int c;
    while ( ( c = getchar() ) == ' ' );
    if ( (c == '.') || (isdigit(c)) ) {
        ungetc(c, stdin);
        scanf("%lf", &yylval);
        return NUMBER;
    }
    return c;
}
```

Figure 4.59: Yacc specification for a more advanced desk calculator.

A reduce/reduce conflict is resolved by choosing the conflicting production listed first in the Yacc specification.

A shift/reduce conflict is resolved in favor of shift. This rule resolves the shift /reduce conflict arising from the dangling-else ambiguity correctly.

Since these default rules may not always be what the compiler writer wants, **Yacc** provides a general mechanism for resolving shift/reduce conflicts. In the declarations portion, we can assign precedences and associativities to terminals. The declaration

'/.left ' + ' '-'

makes + and - be of the same precedence and be left associative. We can declare an operator to be right associative by writing

'/.right

and we can force an operator to be a nonassociative binary operator (i.e., two occurrences of the operator cannot be combined at all) by writing

°/0 nonassoc ' < '

The tokens are given precedences in the order in which they appear in the declarations part, lowest first. Tokens in the same declaration have the same precedence. Thus, the declaration

'/.right UMINUS

in Fig. **4.59** gives the token **UMINUS** a precedence level higher than that of the five preceding terminals.

Yacc resolves shift/reduce conflicts by attaching a precedence and associa-tivity to each production involved in a conflict, as well as to each terminal involved in a conflict. If it must choose between shifting input symbol *a* and re-ducing by production *A* ->• a, **Yacc** reduces if the precedence of the production is greater than that of a, or if the precedences are the same and the associativity of the production is **left.** Otherwise, shift is the chosen action.

Normally, the precedence of a production is taken to be the same as that of its rightmost terminal. This is the sensible decision in most cases. For example, given productions

E ->• E + E | E+E

we would prefer to reduce by E -^-E+E with lookahead +, because the + in the body has the same precedence as the lookahead, but is left associative. With lookahead *, we would prefer to shift, because the lookahead has higher precedence than the + in the production.

In those situations where the rightmost terminal does not supply the proper precedence to a production, we can force a precedence by appending to a pro-duction the tag

°/.prec (terminal)

The precedence and associativity of the production will then be the same as that of the terminal, which presumably is defined in the declaration section. **Yacc** does not report shift/reduce conflicts that are resolved using this precedence and associativity mechanism.

This "terminal" can be a placeholder, like UMINUS in Fig. 4.59; this termi-nal is not returned by the lexical analyzer, but is declared solely to define a precedence for a production. In Fig. 4.59, the declaration

•/.**right** UMINUS

assigns to the token UMINUS a precedence that is higher than that of * and /. In the translation rules part, the tag:

⁰ /.prec UMINUS

at the end of the production

expr : '-' expr

SSM INSTITUTE OF ENGINEERING AND TECHNOLOGY

Dindigul- Palani Highway, Dindigul – 624 002.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

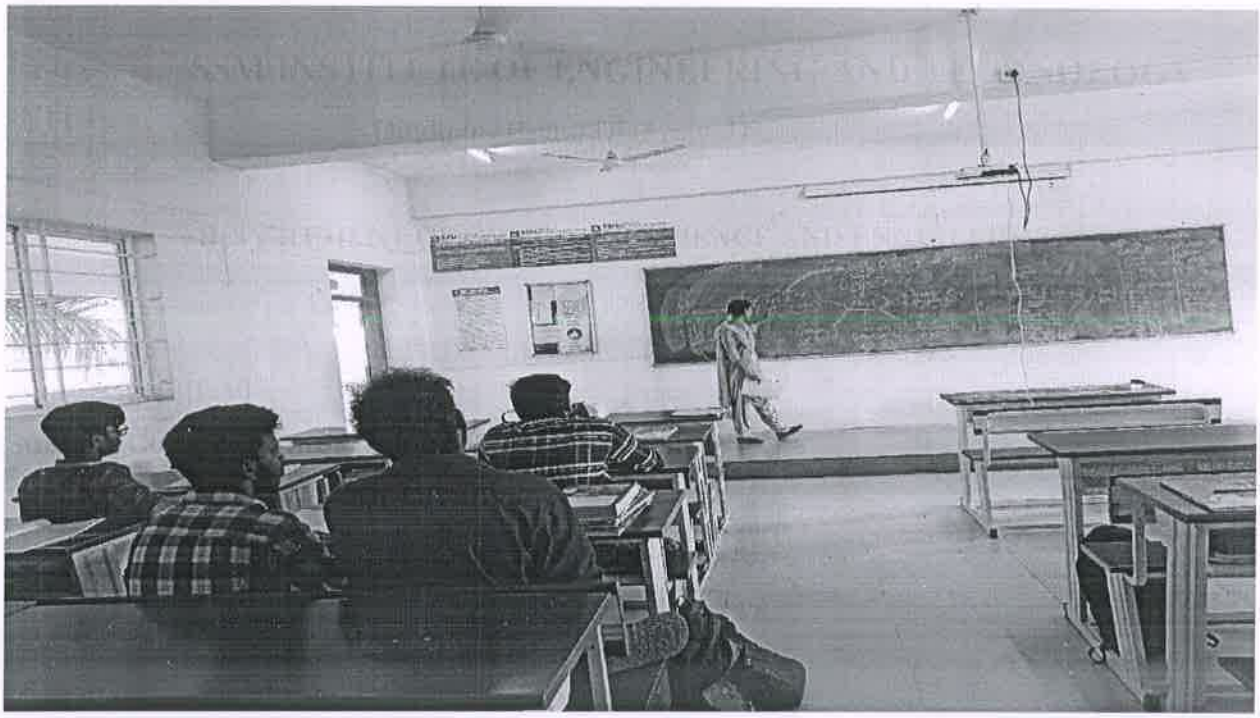
Date: 02.05.2023

Year/Sem: III/VI

Subject Code & Name: CS8602- COMPILER DESIGN

Faculty Handling: N. Anu Lavanya

SEMINAR



FACULTY INCHARGE

HOD/CSE

SSM INSTITUTE OF ENGINEERING AND TECHNOLOGY, Dindigul-624002

Department of Computer Science and Engineering

Internal Test 1

Sub.Code: CS8602 Sub.Name: Compiler Design Year / Sem: III / VI

Sl. No	Reg.NO	Name of the student	IT 1 Original Mark	IT 1 Retest Mark
1	922120104001	ABINAYA G	86	86
2	922120104002	ABINAYA T	86	86
3	922120104003	AKALYA A N	96	96
4	922120104004	ANU P	94	94
5	922120104005	ASHOK KUMAR M	64	56
6	922120104006	ASMETAA G Y	96	96
7	922120104007	BALASURYA R	38	70
8	922120104008	BHOOMIKA R	94	94
9	922120104009	DINESH RAJA E	50	64
10	922120104010	EVANS ABRAHAM J	50	72
11	922120104011	HAREESWARAN S	58	60
12	922120104012	HARINI S	78	78
13	922120104013	JEEVA J	78	78
14	922120104014	JEYASHREE S	70	70
15	922120104015	JOHANS PRAVEEN S	16	70
16	922120104016	JOSEPHINE JESILA M	90	90
17	922120104018	KARUNYA M D	84	84
18	922120104019	KARUPPAIYA M	38	64
19	922120104022	LOGATHARANI S	86	86
20	922120104023	LOGESHWARI P	96	96
21	922120104024	LOKESH G	58	76
22	922120104025	MADHESH KUMAR D	58	78
23	922120104026	MOHAMED ARSATH M	38	70
24	922120104027	MOHAMED FAZIL J	48	60
25	922120104028	MOHAMED HADHI S	58	64
26	922120104029	MUGILAN M	62	62
27	922120104030	PARTHI PRASATH N	64	64
28	922120104031	PIRUTHVI RAMANA V	66	70
29	922120104032	POOJA M	96	96
30	922120104033	PRADEEP V	96	96
31	922120104034	PRAKASH S	66	66
32	922120104035	PREETHIGA M	90	90
33	922120104036	PRETHEEBA U	96	96
34	922120104037	REENA M	94	94
35	922120104038	SABARIKRISHNAN R	74	74
36	922120104039	SAKTHI VIGNESHWARAN B	96	96
37	922120104040	SANJAY PANDI M	38	68
38	922120104041	SANJEEV SARAVANAN S	10	60

39	922120104042	SANTHIYADHARSHINI S	92	92
40	922120104043	SANTHOSH R	50	80
41	922120104044	SARAN PANDIAN S	54	66
42	922120104045	SATHEESH KUMAR K	76	76
43	922120104046	SHALINI J	94	94
44	922120104047	SHARMILA S	96	96
45	922120104049	SINDHUJA INFANT A	86	86
46	922120104050	SIVA SHANTHANA BHARATHI M	48	70
47	922120104051	SIVASUNDAR V	62	70
48	922120104052	SOUNDHARYA DEVI M	94	94
49	922120104053	SRIDHARAN S	52	60
50	922120104054	SRIRAM J	74	74
51	922120104055	SRIRAM S	82	82
52	922120104056	SRIRAM PRASATH L	AB	66
53	922120104057	SUBBIRAMANI R	44	64
54	922120104058	SUBHA S	96	96
55	922120104059	TAMIL ARASAN K	60	80
56	922120104060	VARSHINI U	96	96
57	922120104061	VINOTH KUMAR A	40	72
58	922120104301	SRIRAM V M	50	60

Faculty In-Charge

HOD/CSE

S S M INSTITUTE OF ENGINEERING AND TECHNOLOGY, DINDIGUL 624 002

Approved by AICTE, Delhi and affiliated to Anna University

Accredited by NAAC

College Code 9221 | INTERNAL TEST I / II / III ✓ | Register Number 922120104015

College Name	SSMIET		
Degree/Branch	BE/CSE	Semester	VI
Subject Code	CS8602	Date & Session	24/3/23 (AN
Subject Title	Compiler Design	No. of Pages Used	12

Question Paper Code		All particulars given are verified	Signature of the Hall Supdt. with date
Chief Superintendent's Signature/Facsimile		Name of the Hall Supdt.	

DO NOT WRITE THE REGISTER NUMBER, COLLEGE CODE AND THE NAME IN ANY OTHER PART OF THE ANSWER BOOK

Date: 24/3/23 Session: FORENOON

Subject Code/Title: CS8602 Compiler Design

Question Paper Code: | No. of Pages used: Twelve

Put a tick mark (✓) for the questions attended (to be ticked by the candidate)

PART - A			PART - B									
Question No.	✓	Marks	Question No.		i ✓	i Marks	ii ✓	ii Marks	iii ✓	iii Marks	Total
1	✓	2	6	a		10					
				b							
2	✓	2	7	a							
				b							
3	✓	2	8	a		10					
				b							
4	✓	1	9	a							
	✓			b							
5		2	10	a		b					
				b							
TOTAL											

Three five

GRAND TOTAL (IN WORDS)

25

35

GRAND TOTAL

For Office Use Only

Declaration by the Examiner : Verified that all the questions attended by the student are valued and the total is found to be correct

Date	Signature of the Examiner	Name of the Examiner
24/3/23		N Anuloongs

SSM INSTITUTE OF ENGINEERING AND TECHNOLOGY

Dindigul- Palani Highway, Dindigul – 624 002.

Department of Computer Science and Engineering

Internal Assessment – I
RETEST

Subject Code and Name: CS8602- Compiler Design

Year / SEM: III/ VI
Date: 31.03.2023 (AN)

Time: 02:00pm – 04:30pm
Max Marks: 50
Reg No:- _____

	Part A	**5 X 2 = 10 marks**
1.	Define tokens, patterns and lexemes with example.	R,CO1
2.	What is a sentinel? What is its usage?	R,CO2
3.	Name the various parts in LEX program. With syntax	A,CO2
4.	Define a context free grammar.	U,CO2
5.	Differentiate between Top Down Parser and Bottom Up Parser.	U,CO3
	Part B	**2X16=32 marks**
6)	Describe various phases of compiler. Write down the output of each phases for the expression. a:=b+c-20	Az,CO1
	(OR)	
7)	Construct the minimized DFA for the RE using tree construction method (a/b)* ab #.	A,CO2
8)	Explain backtracking, left recursion and left factoring. Eliminate left recursion for the following grammar. A->ABd|Aa|a, B->Be|b. Eliminate left recursion for the following grammar. E->E+T|T, T->T*F|F, F->(E)|id	A,CO3
	(OR)	
9)	Describe recursive descent parsing and generate the procedure for the following. E->T+E|T T->V*T|V V->id	A,CO3
	Part C	**1X8= 8 marks**
10)	Considering the alphabet ∑={0,1}. Construct a NFA using Thompson construction that is able to recognize the sentences generated by the regular expression.(1*01*0)*1*	A,CO2

Faculty Member

Course Co-ordinator

HoD/CSE 23/3/23

SSM INSTITUTE OF ENGINEERING AND TECHNOLOGY, DINDIGUL-624 002

Approved by AICTE, Delhi and affiliated to Anna University
Accredited by NAAC

College Code	9221	INTERNAL TEST I / II / III	Register Number	922120104008

College Name	SSMIET		
Degree/Branch	BE/CSE	Semester	VI
Subject Code	CS8602	Date & Session	17.03.2023 AN
Subject Title	Compiler Design	No. of Pages Used	22

Question Paper Code		All particulars given are verified	Signature of the Hall Supdt. with date
Chief Superintendent's Signature/Facsimile		Name of the Hall Supdt.	U. Karthick

DO NOT WRITE THE REGISTER NUMBER, COLLEGE CODE AND THE NAME IN ANY OTHER PART OF THE ANSWER BOOK

Date: 17.03.2023 Session: AN

Subject Code/Title: CS8602 Compiler Design

Question Paper Code: No. of Pages used: 22

Put a tick mark (✓) for the questions attended (to be ticked by the candidate)

PART - A			PART - B									
Question No.	✓	Marks	Question No.		i ✓	i Marks	ii ✓	ii Marks	iii ✓	iii Marks	Total
1	✓	2	6	a							
				b							
2	✓	2	7	a	✓	13					
				b							
3	✓	2	8	a							
				b							
4	✓	2	9	a	✓	15					
				b							
5	✓	2	10	a	✓	8					
				b							
TOTAL											

GRAND TOTAL (IN WORDS): Four Eight

48/50

48

GRAND TOTAL

For Office Use Only

Declaration by the Examiner : Verified that all the questions attended by the student are valued and the total is found to be correct

Date	Signature of the Examiner	Name of the Examiner
18/3/23		N. Anu Lovenya

SSM INSTITUTE OF ENGINEERING AND TECHNOLOGY, DINDIGUL-624 002

Approved by AICTE, Delhi and affiliated to Anna University
Accredited by NAAC

College Code	9221	INTERNAL TEST I / II / III	Register Number	922120104008

College Name	SSMIET		
Degree/Branch	BE/CSE	Semester	VI
Subject Code	CS8602	Date & Session	17.03.2023 AN
Subject Title	Compiler Design	No. of Pages Used	22

Question Paper Code		All particulars given are verified	Signature of the Hall Supdt. with date
Chief Superintendent's Signature/Facsimile		Name of the Hall Supdt.	U. Karthick

DO NOT WRITE THE REGISTER NUMBER, COLLEGE CODE AND THE NAME IN ANY OTHER PART OF THE ANSWER BOOK

Date: 17.03.2023 Session: AN

Subject Code/Title: CS8602 Compiler Design

Question Paper Code: No. of Pages used: 22

Put a tick mark (✓) for the questions attended (to be ticked by the candidate)

PART - A			PART - B									
Question No.	✓	Marks	Question No.		i ✓	i Marks	ii ✓	ii Marks	iii ✓	iii Marks	Total
1	✓	2	6	a							
				b							
2	✓	2	7	a	✓	15					
				b							
3	✓	2	8	a							
				b							
4	✓	2	9	a	✓	15					
				b							
5	✓	2	10	a	✓	8					
				b							
TOTAL											

GRAND TOTAL (IN WORDS): Four Eight

48/50

48

GRAND TOTAL

For Office Use Only

Declaration by the Examiner : Verified that all the questions attended by the student are valued and the total is found to be correct

Date	Signature of the Examiner	Name of the Examiner
18/3/23		N. Anu Lavanya

SSM INSTITUTE OF ENGINEERING AND TECHNOLOGY, DINDIGUL-624 002

Approved by AICTE, Delhi and affiliated to Anna University
Accredited by NAAC

College Code	9221	INTERNAL TEST I / II / III	Register Number	922120104012

College Name	SSM INSTITUTE OF ENGINEERING AND TECHNOLOGY		
Degree/Branch	BE / CSE	Semester	VI
Subject Code	CS8602	Date & Session	17.03.22 AN
Subject Title	COMPILER DESIGN	No. of Pages Used	14

Question Paper Code		All particulars given are verified	Signature of the Hall Supdt. with date 17/3/23
Chief Superintendent's Signature/Facsimile		Name of the Hall Supdt.	Murugan. G.

DO NOT WRITE THE REGISTER NUMBER, COLLEGE CODE AND THE NAME IN ANY OTHER PART OF THE ANSWER BOOK

Date: 17.03.22 Session AN

Subject Code/Title CS8602 COMPILER DESIGN

Question Paper Code ..... No. of Pages used fourteen

Put a tick mark (✓) for the questions attended (to be ticked by the candidate)

PART - A			PART - B									
Question No.	✓	Marks	Question No.		i ✓	i Marks	ii ✓	ii Marks	iii ✓	iii Marks	Total
1	✓	1	6	a							
				b							
2	✓	2	7	a	✓	V-					
				b							
3			8	a	✓	9+3					
				b							
4	✓	2	9	a							
				b							
5	✓	1	10	a	✓	6					
				b							
TOTAL											

Three nine

GRAND TOTAL (IN WORDS)

39 39/50

GRAND TOTAL

For Office Use Only

Declaration by the Examiner : Verified that all the questions attended by the student are valued and the total is found to be correct

Date	Signature of the Examiner	Name of the Examiner
18/3/23		N. Anu Lavanya

SSM INSTITUTE OF ENGINEERING AND TECHNOLOGY, DINDIGUL-624 002

Approved by AICTE, Delhi and affiliated to Anna University
Accredited by NAAC

College Code	9221	INTERNAL TEST I / II / III ✓	Register Number	922120104075

College Name	SSMIET		
Degree/Branch	B.E/CSE	Semester	VI
Subject Code	CS8602	Date & Session	17/03/23/AN
Subject Title	Compiler Design	No. of Pages Used	5

Question Paper Code		All particulars given are verified	Signature of the Hall Supdt. with date
Chief Superintendent's Signature/Facsimile		Name of the Hall Supdt.	U.V. Karthick

DO NOT WRITE THE REGISTER NUMBER, COLLEGE CODE AND THE NAME IN ANY OTHER PART OF THE ANSWER BOOK

Date: 17/03/23 Session: AFTERNOON

Subject Code/Title: CS8602 compiler Design

Question Paper Code: Number of Pages used: five

Put a tick mark (✓) for the questions attended (to be ticked by the candidate)

PART - A			PART - B									
Question No.	✓	Marks	Question No.		i ✓	i Marks	ii ✓	ii Marks	iii ✓	iii Marks	Total
1	✓	1	6	a	✓	5	✓				
				b							
2			7	a							
				b							
3			8	a							
				b							
4	✓		9	a							
	✓			b							
5	✓	2	10	a							
				b							
TOTAL											

Eight

GRAND TOTAL (IN WORDS)

08/50

8

GRAND TOTAL

For Office Use Only

Declaration by the Examiner : Verified that all the questions attended by the student are valued and the total is found to be correct

Date	Signature of the Examiner	Name of the Examiner
18/3/23		N. Anuloga(?)

SSM INSTITUTE OF ENGINEERING AND TECHNOLOGY

Dindigul- Palani Highway, Dindigul – 624 002.

Department of Computer Science and Engineering

Internal Assessment – I

Subject Code and Name: CS8602- Compiler Design

Year / SEM: III/ VI
Date: 17.03.2023 (AN)

Time: 02:00pm – 03:30pm
Max Marks: 50
Reg No:- _____

	Part A	**5 X 2 = 10 marks**
1.	Define tokens, patterns and lexemes with example.	R,CO1
2.	Draw a transition diagram to represent relational operators.	R,CO2
3.	Name the various parts in LEX program. With syntax	A,CO2
4.	What is an ambiguous grammar? Prove E->E+E|E*E|id	U,CO2
5.	Differentiate between Top Down Parser and Bottom Up Parser.	U,CO3
	Part B	**2X16=32 marks**
6)	Describe various phases of compiler. Write down the output of each phases for the expression. a:=b+c-20	Az,CO1
	(OR)	
7)	Convert the given NFA into its equivalent DFA. 	A,CO2
8)	Explain backtracking, left recursion and left factoring. Eliminate left recursion for the following grammar. A->ABd|Aa|a, B->Be|b. Eliminate left recursion for the following grammar. E->E+T, T->T*F|F, F->(E)|id	A,CO3
	(OR)	
9)	Describe recursive descent parsing and generate the procedure for the following. E->E+T|T T->T*F|F F->(E)|id	A,CO3
	Part C	**1X8= 8 marks**
10)	Considering the alphabet ∑={0,1}. Construct a NFA using Thompson construction that is able to recognize the sentences generated by the regular expression.(1*01*0)*1*	A,CO2

Faculty Member

Course Co-ordinator

HoD

SSM INSTITUTE OF ENGINEERING AND TECHNOLOGY, Dindigul-624002

Department of Computer Science and Engineering

Internal Test II

Sub.Code: CS8602 Sub.Name: Compiler Design Year / Sem: III / VI

Sl. No	Reg.NO	Name of the student	IT II	IT II
			Original Mark	Retest Mark
1	922120104001	ABINAYA G	58	76
2	922120104002	ABINAYA T	60	72
3	922120104003	AKALYA A N	90	90
4	922120104004	ANU P	AB	86
5	922120104005	ASHOK KUMAR M	60	72
6	922120104006	ASMETAA G Y	64	74
7	922120104007	BALASURYA R	56	72
8	922120104008	BHOOMIKA R	100	100
9	922120104009	DINESH RAJA E	32	72
10	922120104010	EVANS ABRAHAM J	52	70
11	922120104011	HAREESWARAN S	38	70
12	922120104012	HARINI S	72	72
13	922120104013	JEEVA J	76	76
14	922120104014	JEYASHREE S	52	74
15	922120104015	JOHANS PRAVEEN S	8	76
16	922120104016	JOSEPHINE JESILA M	60	70
17	922120104018	KARUNYA M D	60	74
18	922120104019	KARUPPAIYA M	34	74
19	922120104022	LOGATHARANI S	48	78
20	922120104023	LOGESHWARI P	50	76
21	922120104024	LOKESH G	92	92
22	922120104025	MADHESH KUMAR D	78	78
23	922120104026	MOHAMED ARSATH M	AB	78
24	922120104027	MOHAMED FAZIL J	32	68
25	922120104028	MOHAMED HADHI S	90	90
26	922120104029	MUGILAN M	2	72
27	922120104030	PARTHI PRASATH N	80	80
28	922120104031	PIRUTHVI RAMANA V	90	90
29	922120104032	POOJA M	100	100
30	922120104033	PRADEEP V	98	98
31	922120104034	PRAKASH S	68	68
32	922120104035	PREETHIGA M	80	80
33	922120104036	PRETHEEBA U	100	100
34	922120104037	REENA M	84	84
35	922120104038	SABARIKRISHNAN R	74	74
36	922120104039	SAKTHI VIGNESHWARAN B	84	84
37	922120104040	SANJAY PANDI M	46	78
38	922120104041	SANJEEV SARAVANAN S	52	70

39	922120104042	SANTHIYADHARSHINI S	70	70
40	922120104043	SANTHOSH R	62	70
41	922120104044	SARAN PANDIAN S	52	70
42	922120104045	SATHEESH KUMAR K	70	70
43	922120104046	SHALINI J	92	92
44	922120104047	SHARMILA S	100	100
45	922120104049	SINDHUJA INFANT A	AB	70
46	922120104050	SIVA SHANTHANA BHARATHI M	64	74
47	922120104051	SIVASUNDAR V	AB	70
48	922120104052	SOUNDHARYA DEVI M	100	100
49	922120104053	SRIDHARAN S	56	70
50	922120104054	SRIRAM J	42	70
51	922120104055	SRIRAM S	48	74
52	922120104056	SRIRAM PRASATH L	70	70
53	922120104057	SUBBIRAMANI R	70	70
54	922120104058	SUBHA S	100	100
55	922120104059	TAMIL ARASAN K	82	82
56	922120104060	VARSHINI U	100	100
57	922120104061	VINOTH KUMAR A	46	74
58	922120104301	SRIRAM V M	20	70

Faculty In-Charge

G.Lib

HOD/CSE

SSM INSTITUTE OF ENGINEERING AND TECHNOLOGY, DINDIGUL-624 002

Approved by AICTE, Delhi and affiliated to Anna University
Accredited by NAAC

College Code	9	2	2	1

INTERNAL TEST I / II / III

Register Number: 9 2 2 1 2 0 1 0 4 0 4 7

College Name	SSMIET		
Degree/Branch	BE/CSE	Semester	6
Subject Code	CS8602	Date & Session	18.04.2023 AN
Subject Title	Compiler Design	No. of Pages Used	

Question Paper Code: | | | | | | |

All particulars given are verified — Signature of the Hall Supdt. with date: 18/4/23

Chief Superintendent's Signature/Facsimile: ADJ

Name of the Hall Supdt.: Murugan. G

DO NOT WRITE THE REGISTER NUMBER, COLLEGE CODE AND THE NAME IN ANY OTHER PART OF THE ANSWER BOOK

Date: 18.04.2023 Session: AN

Subject Code/Title: CS8602 Compiler Design

Question Paper Code: | | | | | | |

No. of Pages used:

Put a tick mark (✓) for the questions attended (to be ticked by the candidate)

PART - A			PART - B									
Question No.	✓	Marks	Question No.		i ✓	i Marks	ii ✓	ii Marks	iii ✓	iii Marks	Total
1	✓	2	6	a	✓	16					
				b							
2	✓	2	7	a							
				b							
3	✓	2	8	a							
				b							
4	✓	2	9	a	✓	16					
				b							
5	✓	2	10	a	✓	16					
				b							
TOTAL											

Five zero

GRAND TOTAL (IN WORDS)

50

Kamila

GRAND TOTAL

For Office Use Only

Declaration by the Examiner : Verified that all the questions attended by the student are valued and the total is found to be correct

Date	Signature of the Examiner	Name of the Examiner
18/4/23		N. ANUSUYA

SSM INSTITUTE OF ENGINEERING AND TECHNOLOGY, DINDIGUL-624 002

Approved by AICTE, Delhi and affiliated to Anna University
Accredited by NAAC

College Code	9221	INTERNAL TEST I / II / III	Register Number	922120104028

College Name	SSMIET		
Degree/Branch	B.E / CSE	Semester	6th
Subject Code	CS8602	Date & Session	18.04.2023 (AN)
Subject Title	Compiler design	No. of Pages Used	

Question Paper Code		All particulars given are verified	Signature of the Hall Supdt. with date 18/4/23
Chief Superintendent's Signature/Facsimile		Name of the Hall Supdt.	Murugan. G.

DO NOT WRITE THE REGISTER NUMBER, COLLEGE CODE AND THE NAME IN ANY OTHER PART OF THE ANSWER BOOK

Date: 18.4.2023 Session: (A.N)

Subject Code/Title: CS8602 compiler design.

Question Paper Code: ..... No. of Pages used: .....

Put a tick mark (✓) for the questions attended (to be ticked by the candidate)

PART - A			PART - B									
Question No.	✓	Marks	Question No.		i ✓	i Marks	ii ✓	ii Marks	iii ✓	iii Marks	Total
1		2	6	a							
				b							
2		2	7	a		8					
				b							
3		1	8	a							
				b							
4		2	9	a		12					
				b							
5		1	10	a		8					
				b							
TOTAL											

Three six

GRAND TOTAL (IN WORDS)

36

Hashi

GRAND TOTAL 36

For Office Use Only

Declaration by the Examiner : Verified that all the questions attended by the student are valued and the total is found to be correct

Date	Signature of the Examiner	Name of the Examiner
19/4/23		N. Anulakshmi

SSM INSTITUTE OF ENGINEERING AND TECHNOLOGY, DINDIGUL-624 002

Approved by AICTE, Delhi and affiliated to Anna University
Accredited by NAAC

College Code	9221	INTERNAL TEST I / II / III	Register Number	922120104029

College Name	SSMIET		
Degree/Branch	BE / CSE	Semester	VI
Subject Code	CS8602	Date & Session	18/4/22 A.N
Subject Title	Compiler Design	No. of Pages Used	

Question Paper Code		All particulars given are verified	Signature of the Hall Supdt. with date 18/4/23
Chief Superintendent's Signature/Facsimile		Name of the Hall Supdt.	Murugan. G.

DO NOT WRITE THE REGISTER NUMBER, COLLEGE CODE AND THE NAME IN ANY OTHER PART OF THE ANSWER BOOK

Date: 18.4.23 Session AN COPY

Subject Code/Title CS8602 Compiler Design.

Question Paper Code ____ No. of Pages used 10

Put a tick mark (✓) for the questions attended (to be ticked by the candidate)

PART - A			PART - B									
Question No.	✓	Marks	Question No.		i ✓	i Marks	ii ✓	ii Marks	iii ✓	iii Marks	Total
1		2	6	a		6					
				b							
2		2	7	a							
				b							
3		1	8	a							
				b							
4			9	a							
				b							
5			10	a							
				b							
TOTAL											

one one

GRAND TOTAL (IN WORDS) ten one

GRAND TOTAL

For Office Use Only

Declaration by the Examiner : Verified that all the questions attended by the student are valued and the total is found to be correct

Date	Signature of the Examiner	Name of the Examiner
19/4/23		N. Anu Lavanya

SSM INSTITUTE OF ENGINEERING AND TECHNOLOGY

Dindigul- Palani Highway, Dindigul – 624 002.

Department of Computer Science and Engineering

Internal Assessment – II
RETEST

Subject Code and Name: CS8602- Compiler Design

Year / SEM: III/ VI
Date: 21.04.2023 (FN)

Time: 09:00am – 10:30am
Max Marks: 50
Reg No:- _____

	Part A	**5 X 2 = 10 marks**
1.	List the error recovery strategies.	A,CO3
2.	What are kernel and non-kernel items?	R,CO3
3.	What do you mean by syntax directed translation	R,CO4
4.	Write the types of intermediate languages.	U,CO4
5.	Define backpatching.	R,CO4
	Part B	**2X16=32 marks**
6)	Describe and apply LR(0) parsing algorithm on the following grammar S->AA, A->aA|b	A,CO3
	(OR)	
7)	Describe and apply CLR(1) & LALR(1) parsing algorithm on the following grammar: E->BB, B->cB|d	A,CO3
8)	Explain the construction of syntax tree for expression with an example.	U,CO4
	(OR)	
9)	What are the various methods of implementing three address statements? Write the quadruples, triple and indirect triples for the expression: (a+b)*(c+d)-(a+b+c).	A,CO4
	Part C	**1X8= 8 marks**
10)	Analyze the difference between Synthesized and Inherited SDT with example.	A,CO4

Faculty Member

Course Co-ordinator

HoD

S S M INSTITUTE OF ENGINEERING AND TECHNOLOGY, DINDIGUL-624 002
Approved by AICTE, Delhi and affiliated to Anna University
Accredited by NAAC

Register Number: 9 2 2 1 2 0 1 0 4 0 2 9

College Code: 9 2 2 1 — INTERNAL TEST I / II / III

College Name	SSMIET		
Degree/Branch	BE/CSE	Semester	VI
Subject Code	CS8602.	Date & Session	21.4.'23 AFN
Subject Title	Compiler Design.	No. of Pages Used	

Question Paper Code		All particulars given are verified	Signature of the Hall Supdt with date
Chief Superintendent's Signature/Facsimile		Name of the Hall Supdt.	

DO NOT WRITE THE REGISTER NUMBER, COLLEGE CODE AND THE NAME IN ANY OTHER PART OF THE ANSWER BOOK

Date: 21.4.23 Session: AN

Subject Code/Title: CS8602 compile design.

Question Paper Code: ____ No. of Pages used: ____

Put a tick mark (✓) for the questions attended (to be ticked by the candidate)

PART - A			PART - B									
Question No.	✓	Marks	Question No.		i ✓	i Marks	ii ✓	ii Marks	iii ✓	iii Marks	Total
1	✓	2	6	a	✓	16					
				b							
2	✓	2	7	a							
				b							
3	✓	2	8	a							
				b							
4	✓	2	9	a	✓	10					
				b							
5	✓	2	10	a							
				b							
TOTAL											

GRAND TOTAL (IN WORDS): Three six

36

GRAND TOTAL

For Office Use Only

Declaration by the Examiner : Verified that all the questions attended by the student are valued and the total is found to be correct

21/4/23 Date	Signature of the Examiner	N. Anu Lavanya Name of the Examiner

SSM INSTITUTE OF ENGINEERING AND TECHNOLOGY, Dindigul-624002

Department of Computer Science and Engineering

Internal Test III

Sub.Code: CS8602 Sub.Name: Compiler Design Year / Sem: III / VI

Sl. No	Reg.NO	Name of the student	IT III Original Mark	IT III Retest Mark
1	922120104001	ABINAYA G	92	92
2	922120104002	ABINAYA T	72	72
3	922120104003	AKALYA A N	92	92
4	922120104004	ANU P	90	90
5	922120104005	ASHOK KUMAR M	62	75
6	922120104006	ASMETAA G Y	48	64
7	922120104007	BALASURYA R	54	76
8	922120104008	BHOOMIKA R	72	72
9	922120104009	DINESH RAJA E	42	77
10	922120104010	EVANS ABRAHAM J	56	76
11	922120104011	HAREESWARAN S	36	80
12	922120104012	HARINI S	64	64
13	922120104013	JEEVA J	54	63
14	922120104014	JEYASHREE S	48	74
15	922120104015	JOHANS PRAVEEN S	76	76
16	922120104016	JOSEPHINE JESILA M	50	63
17	922120104018	KARUNYA M D	54	65
18	922120104019	KARUPPAIYA M	70	75
19	922120104022	LOGATHARANI S	74	74
20	922120104023	LOGESHWARI P	84	84
21	922120104024	LOKESH G	90	90
22	922120104025	MADHESH KUMAR D	AB	65
23	922120104026	MOHAMED ARSATH M	44	62
24	922120104027	MOHAMED FAZIL J	66	83
25	922120104028	MOHAMED HADHI S	86	86
26	922120104029	MUGILAN M	48	78
27	922120104030	PARTHI PRASATH N	62	79
28	922120104031	PIRUTHVI RAMANA V	60	60
29	922120104032	POOJA M	100	100
30	922120104033	PRADEEP V	94	94
31	922120104034	PRAKASH S	74	79
32	922120104035	PREETHIGA M	56	66
33	922120104036	PRETHEEBA U	100	100
34	922120104037	REENA M	28	62
35	922120104038	SABARIKRISHNAN R	88	88
36	922120104039	SAKTHI VIGNESHWARAN B	OD	75
37	922120104040	SANJAY PANDI M	54	75
38	922120104041	SANJEEV SARAVANAN S	34	82
39	922120104042	SANTHIYADHARSHINI S	92	92

40	922120104043	SANTHOSH R	84	84
41	922120104044	SARAN PANDIAN S	28	78
42	922120104045	SATHEESH KUMAR K	70	70
43	922120104046	SHALINI J	AB	75
44	922120104047	SHARMILA S	96	96
45	922120104049	SINDHUJA INFANT A	100	100
46	922120104050	SIVA SHANTHANA BHARATHI M	AB	66
47	922120104051	SIVASUNDAR V	62	75
48	922120104052	SOUNDHARYA DEVI M	96	96
49	922120104053	SRIDHARAN S	52	77
50	922120104054	SRIRAM J	OD	69
51	922120104055	SRIRAM S	40	65
52	922120104056	SRIRAM PRASATH L	OD	75
53	922120104057	SUBBIRAMANI R	52	76
54	922120104058	SUBHA S	98	98
55	922120104059	TAMIL ARASAN K	18	62
56	922120104060	VARSHINI U	100	100
57	922120104061	VINOTH KUMAR A	AB	76
58	922120104301	SRIRAM V M	36	82

Faculty In-Charge

HOD/CSE

SSM INSTITUTE OF ENGINEERING AND TECHNOLOGY

Dindigul- Palani Highway, Dindigul – 624 002.

Department of Computer Science and Engineering

Internal Assessment – III

Subject Code and Name: CS8602- Compiler Design

Year / SEM: III/ VI
Date: 19.04.2023 (AN)

Time: 10:30am – 12:00pm
Max Marks: 50
Reg No:- _____

	Part A	**5 X 2 = 10 marks**
1.	Distinguish between static and dynamic storage allocation	U,CO4
2.	Define the non-local data on stack	R,CO4
3.	What is DAG? Point out advantages of DAG.	A,CO5
4.	Define algebraic transformations. Give an example	U,CO5
5.	State the use of machine Idioms	R,CO5
	Part B	**2X16=32 marks**
6)	Compare the stack and heap allocation memory in detail with suitable examples.	A,CO4
	(OR)	
7)	Explain in detail about the various issues in code generation with examples	A,CO4
8)	Explain briefly about the Peephole optimization with suitable examples.	U,CO5
	(OR)	
9)	Explain briefly about the principal sources of optimization	U,CO5
	Part C	**1X8= 8 marks**
10)	Explain any two techniques in Loop Optimization.	U,CO5

Faculty Member

Course Co-ordinator

HoD 12/5/23

922120104054.
Sriram. J

Compiler Design.
Retest - III

3½

6. Peep-hole optimization:

If we apply the statement by statement code generation strategy then the generated target code may contain many redudant instructions. The quantity of such code is very poor. To optimize such a target code certain transformations need to be applied on the target code. These transformations ultimately will result in getting the significant improvement over the running time or space requirement of the target program.

SSM INSTITUTE OF ENGINEERING AND TECHNOLOGY, DINDIGUL-624 002

Approved by AICTE, Delhi and affiliated to Anna University
Accredited by NAAC

College Code	9221	INTERNAL TEST I / II / III	Register Number	922120104049

College Name	SSMIET		
Degree/Branch	B.E CSE	Semester	VI
Subject Code	CS8602	Date & Session	19/05/23 / FN
Subject Title	Compiler Design	No. of Pages Used	13

Question Paper Code		All particulars given are verified
Chief Superintendent's Signature/Facsimile		Signature of the Hall Supdt. with date
		S. Soundarlingam. Name of the Hall Supdt.

DO NOT WRITE THE REGISTER NUMBER, COLLEGE CODE AND THE NAME IN ANY OTHER PART OF THE ANSWER BOOK

Date: 19/05/23 Session: FN

Subject Code/Title: CS8602 Compiler Design

Question Paper Code: No. of Pages used:

Put a tick mark (✓) for the questions attended (to be ticked by the candidate)

PART - A			PART - B									
Question No.	✓	Marks	Question No.		i ✓	i Marks	ii ✓	ii Marks	iii ✓	iii Marks	Total
1	✓	2	6	a	✓	16					
				b							
2	✓	2	7	a							
				b							
3	✓	2	8	a	✓	16					
				b							
4	✓	2	9	a							
				b							
5	✓	2	10	a	✓	8					
				b							
TOTAL											

Five zero

GRAND TOTAL (IN WORDS)

50

GRAND TOTAL

For Office Use Only

Declaration by the Examiner : Verified that all the questions attended by the student are valued and the total is found to be correct

Date	Signature of the Examiner	Name of the Examiner
19/5/23		N. Anu Lavanya

SSM INSTITUTE OF ENGINEERING AND TECHNOLOGY, DINDIGUL-624 002

Approved by AICTE, Delhi and affiliated to Anna University
Accredited by NAAC

College Code	9221	INTERNAL TEST I / II / III	Register Number	922120104019

College Name	SSMIET		
Degree/Branch	BE/CSE	Semester	VI
Subject Code	CS8602	Date & Session	19.5.23 (FN)
Subject Title	Compiler Design	No. of Pages Used	

Question Paper Code	All particulars given are verified — N.J.D 19/5/23 Signature of the Hall Supdt. with date
Chief Superintendent's Signature/Facsimile	N.J.DIVYA Name of the Hall Supdt.

DO NOT WRITE THE REGISTER NUMBER, COLLEGE CODE AND THE NAME IN ANY OTHER PART OF THE ANSWER BOOK

Date: 19.5.23 Session: (FN)

Subject Code/Title: CS8602 Compiler Design

Question Paper Code: ____ No. of Pages used: ____

Put a tick mark (√) for the questions attended (to be ticked by the candidate)

PART - A			PART - B									
Question No.	√	Marks	Question No.		i √	i Marks	ii √	ii Marks	iii √	iii Marks	Total
1	√	2	6	a	√	10					
				b							
2			7	a							
				b							
3	√	2	8	a	√	15					
				b							
4	√	2	9	a							
				b							
5	√	2	10	a		2					
				b							
TOTAL											

THIRTY FIVE

GRAND TOTAL (IN WORDS)

35

GRAND TOTAL

For Office Use Only

Declaration by the Examiner : Verified that all the questions attended by the student are valued and the total is found to be correct

Date	Signature of the Examiner	Name of the Examiner
19/5/23		N. Anu Lavanya

S S M INSTITUTE OF ENGINEERING AND TECHNOLOGY, DINDIGUL-624 002

Approved by AICTE, Delhi and affiliated to Anna University
Accredited by NAAC

College Code	9221	INTERNAL TEST I / II / III	Register Number	922120104059

College Name	SSMIET		
Degree/Branch	BE CSE	Semester	06
Subject Code	CS8602	Date & Session	19.05.23
Subject Title	Compiler Design	No. of Pages Used	

Question Paper Code		All particulars given are verified	Signature of the Hall Supdt. with date
Chief Superintendent's Signature/Facsimile		Name of the Hall Supdt.	S. Soundarlingam

DO NOT WRITE THE REGISTER NUMBER, COLLEGE CODE AND THE NAME IN ANY OTHER PART OF THE ANSWER BOOK

Date: 19.05.23 Session: FN

Subject Code/Title: CS8602 Compiler Design

Question Paper Code: No. of Pages used:

Put a tick mark (✓) for the questions attended (to be ticked by the candidate)

PART - A			PART - B									
Question No.	✓	Marks	Question No.		i ✓	i Marks	ii ✓	ii Marks	iii ✓	iii Marks	Total
1			6	a							
				b							
2			7	a							
				b							
3			8	a		2					
				b							
4			9	a							
				b							
5			10	a		7					
				b							
TOTAL											

GRAND TOTAL (IN WORDS): nine

GRAND TOTAL: 9

For Office Use Only

Declaration by the Examiner : Verified that all the questions attended by the student are valued and the total is found to be correct

Date	Signature of the Examiner	Name of the Examiner
19/5/23		N. Anu Lavanya

Name of the Faculty : N. Anu Lavanya

Designation : AP Department CSE

Branch : B.E. CSE Section

Semester : VI ODD / EVEN Academic Year: 2022 - 2023

Subject Code : CS8602 Title: Compiler Design

Syllabus Coverage

	I Internal Test	II Internal Test	III Internal Test
No. of Units covered	1 1/2	1 1/2 (3)	5
Signature of the Faculty with date	9/3/23	15/4/23	6/5/23
Signature of the HoD with date	G. Lalitha 9/3/23	G. Lalitha 15/4/23	G. Lalitha 6/5/23
Principal			

DETAILS OF COMPLETION OF SYLLABUS

S. No.	Unit	Planned Date	Planned Hrs.	Topics to be covered	Executed Date	Executed Hrs.	Remarks
1.		6/2/23	6	Structure of a Compiler	7/2/23	5	
2		8/2/23	2	Lexical Analysis	8/2/23	4	
3		8/2/23	5	Role of Lexical Analyzer	8/2/23	5	
4		13/2/23	6	Input Buffering	13/2/23	6	
5	I	15/2/23	5	Recognition of Tokens	15/2/23	5	
6		15/2/23	2	Specification & "	15/2/23	2	
7		18/2/23	5	Lex-finite Automata	18/2/23	5	
8		20/2/23	6	RE to Automata	18/2/23	5	
9		22/2/23	2	Minimizing DFA	4/3/23	5	
1.		28/2/23	5	Role of parser	6/3/23	5	
2		27/2/23	6	Grammars	6/3/23	5	
3		1/3/23	2	Error handling	6/3/23	6	
4		1/3/23	5	Context - free grammars	6/3/23	6	
5		4/3/23	5	Writing a grammar	7/3/23	2	
6	II	6/3/23	6	Top down parsing	7/3/23	4	
7		8/3/23	5	General strategies Recursive	7/3/23	5	
8		13/3/23	6	Descent parser predictive parser	8/3/23	5	
9		15/3/23	2,5	LL(1) parser-Shift Reduce parser	9/3/23	4	
10		18/3/23	5	LR parser-LR(0) Item construction of SLR parsing table	9/3/23	7	
11		20/3/23	6	Introduction to LALR parser	18/3/23	5	
12		25/3/23	5	Error handling & Recovery in Syntax Analyzer	20/3/23	6	
		25/3/23	6	YACC			

HoD

Principal

DETAILS OF COMPLETION OF SYLLABUS

S. No.	Unit	Planned Date	Planned Hrs.	Topics to be covered	Executed Date	Executed Hrs.	Remarks
1.		25/3/23	7	Syntax Directed Definition	29/3/23	5	
2		27/3/23	6	Evaluation order for SDD	29/3/23	5	
3		29/3/23	2,5	Evaluation orders for SDD	28/3/23	5	
4	III	1/4/23	5	Intermediate language Syntax	1/4/23	6	
5		3/4/23	6	Three address code	5/4/23	5	
6		5/4/23	2	Types & Declaration	5/4/23	7	
7		5/4/23	4	Translation & Expression	12/4/23	5	
8		8/4/23	1	Type checking	12/4/23	7	
9		8/4/23	2	CBS: parser Generator.	15/4/23	3	
1.		8/4/23	3,4	Storage organisation	15/4/23	7	
2		8/4/23	5	Static Vs Dynamic storage	19/4/23	5	
3		8/4/23	6	Stack Allocation Space	19/4/23	7	
4		8/4/23	7	Local & Non-local data	25/4/23	1	
5	IV	10/4/23	6	Access to Non-local data on the stack	25/4/23	2	
6		12/4/23	2	Heap management	26/4/23	5	
7		12/4/23	5	Issues in Code generation	26/4/23	7	
8		15/4/23	5	Design of a simple Code	28/4/23	1	
1.		17/4/23	6	Principal Sources of optimization	28/4/23	2	
2		19/4/23	5	Peep-hole optimization	29/4/23	3	
3		24/4/23	6	DAG	29/4/23	7	

HoD

Principal

DETAILS OF COMPLETION OF SYLLABUS

S. No.	Unit	Planned Date	Planned Hrs.	Topics to be covered	Executed Date	Executed Hrs.	Remarks
A		26/4/23	2	Types of VM & their implementations	2/5/23	1,2	
5		26/4/23	5	Optimization of Basic blocks	3/5/23	5	
6		29/4/23	5	Global data flow analysis	3/5/23	7	
7	I Unit	3/5/23	2,5	Efficient data flow Alg	6/5/23	5	
8		6/5/23	5	Question bank Discussed	9/5/23	1,2	
9		8/5/23	6	Previous year university QP	13/5/23	1,2	
10		10/5/23	2,5	Unit wise important pb solved.	11/5/23	5	
1.	1/2	Expla		Lexical Analyser	8.3	1,2	
2	14/2/23	1b		Symbol Table	14.3	1,2	
3	21/2 25/2	2		LA using lex Tool	24.3	1,2	
4	7/3 14/3	3		Arithmetic Calculator	28.3	1,2	
5	21/3 25/3	4		Three address code	11.4	1,2	
6	28/3 11/4	5		Simple code optimization	18.4	1,2	
7	18/4 25/4	6		Backend of Compiler	21.4	1,2	
8	9/5	CBS.		LA to i/p text file & Count	23.5	1	
9	9/5	1,2		Model lab	23.5	2,3,4	

HoD

Principal

SSM INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Approved by AICTE, New Delhi / Affiliated to Anna University, Chennai / Accredited by NAAC)
Dindigul – Palani Highway, Dindigul – 624 002

COURSE PLAN

Academic Year	2022-2023	Duration of the semester as per academic calendar	From : 06.02.2023 To : 12.05.2023	
	Name of the Department	COMPUTER SCIENCE AND ENGINEERING	Semester	VI
	Subject Code & Name	CS8602 - Compiler Design	Total Hours as per syllabus	45 30
	Name of the Faculty with designation & department	Ms N. ANU LAVANYA Assistant Professor/ Computer Science engineering		
Regular / ~~Elective~~	Regular	Department	Parent dept.	Other dept.
		Subject	✓	--
Prerequisites Knowledge		Knowledge in Theory of Computation		
Duration		One semester	Credit Units :	4
Class / Laboratory Schedule		3 0 2 4		[L T P C]
Curriculum gap (Content beyond syllabus / Hands-on training / Industrial Visit / Expert Lecture)		CBS: Parser Generator	Mapping of POs and PSOs PO: 1,2,12 PSO: 1,2	

Course Objective:

Students are able

- To learn the various phases of compiler.
- To learn the various parsing techniques.
- To understand intermediate code generation and run-time environment.
- To learn to implement front-end of the compiler.
- To learn to implement code generator.

CO	Course Outcome(s)	Program Outcome(s)
CS8602.1	Understand the different phases of compiler.	PO 01 # Engineering Knowledge PO 02 # Problem analysis PO 03 # Design/development of solutions PO 04 # Conduct investigations of complex problems PO 05 # Modern tool usage PO 06 # The engineer and society PO 07 # Environment and sustainability PO 08 # Ethics PO 09 # Individual and team work PO 10 # Communication PO 11# Project management and finance PO 12# Life-long learning
CS8602.2	Design a lexical analyzer for a sample language. Design and implement a scanner and a parser using LEX and YACC tools	
CS8602.3	Apply different parsing algorithms to develop the parsers for a given grammar.	
CS8602.4	Understand syntax-directed translation and run-time environment.	
CS8602.5	Learn to implement code optimization techniques and a simple code generator.	

S. No.	Planned		Cumulative Hours	Topics to be covered	Text / Ref. Book	Page Nos.	Executed Date	Remarks
	Date	Hour						
UNIT – 1 INTRODUCTION TO COMPILERS								
1	6/2/23	6	1	Structure of a compiler	T1	4	7/2/23	
2	8/2/23	2	2	Lexical Analysis	T1	109	8/2/23	
3	8/2/23	5	3	Role of Lexical Analyzer	T1	115	8/2/23	
4	13/2/23	6	4	Input Buffering	T1	116	13/2/23	
5	15/2/23	2	5	Specification of Tokens	T1	128	15/2/23	
6	15/2/23	5	6	Recognition of Tokens	T1	140	15/2/23	
7	18/2/23	5	7	Lex – Finite Automata	T1	147	18/2/23	
8	20/2/23	6	8	Regular Expressions to Automata	T1	152	18/2/23	
9	22/2/23	2	9	Minimizing DFA	T1	180	4/3/23	
UNIT – 2 SYNTAX ANALYSIS								
1	22/2/23	5	10	Role of Parser	T1	192	6/3/23	
2	27/2/23	6	11	Grammars	T1	194	6/3/23	
3	1/3/23	2	12	Error Handling	T1	194	6/3/23	
4	1/3/23	5	13	Context-free grammars	T1	197	6/3/23	
5	4/3/23	5	14	Writing a grammar	T1	198	7/3/23	
6	6/3/23	6	15	Top Down Parsing	T1	217	7/3/23	
7	8/3/23 13/3/23	5 6	17	General Strategies Recursive Descent Parser Predictive Parser	T1	219	7/3/23	
8	15/3/23	2,5	19	LL(1) Parser-Shift Reduce Parser	T1	222	8/3/23	
9	18/3/23	5	20	LR Parser-LR (0)Item Construction of SLR Parsing Table	T1	241	9/3/23	
10	20/3/23	6	21	Introduction to LALR Parser	T1	266	9/3/23	
11	25/3/23	5	22	Error Handling and Recovery in Syntax Analyzer		295	18/3/23	
12	25/3/23	6	23	YACC	T1	287	20/3/23	
UNIT – 3 INTERMEDIATE CODE GENERATION								
1	25/3/23	7	24	Syntax Directed Definitions	T1	304	29/3/23	
2	27/3/23	6	25	Evaluation Orders for Syntax Directed Definitions	T1	306	29/3/23	
3	29/3/23	2,5	27	Evaluation Orders for Syntax Directed Definitions	T1	310	29/3/23	
4	1/4/23	5	28	Intermediate Languages: Syntax Tree	T1	357	1/4/23	
5	3/4/23	6	29	Three Address Code,	T1	363	5/4/23	
6	5/4/23	2	30	Types and Declarations	T1	370	5/4/23	
7	5/4/23	4	31	Translation of Expressions	T1	378	12/4/23	
8	8/4/23	1	32	Type Checking	T1	386	12/4/23	
9	8/4/23	2	33	CBS: Parser Generator	Ref Material		15/4/23	
UNIT – 4 RUN-TIME ENVIRONMENT AND CODE GENERATION								
1	8/4/23	3,4	35	Storage Organization	T1	427	15/4/23	

2	8/4/23	5	36	Static Vs, Dynamic Storage Allocation	T1	429	19/4/23
3	8/4/23	6	37	Stack Allocation Space	T1	430	19/4/23
4	8/4/23	7	38	Local and Non – Local data	T1	441	25/4/23
5	10/4/23	6	39	Access to Non-local Data on the Stack	T1	442	25/4/23
6	12/4/23	2	40	Heap Management	T1	452	26/4/23
7	12/4/23	5	41	Issues in Code Generation	T1	506	26/4/23
8	15/4/23	5	42	Design of a simple Code Generator.	T1	542	28/4/23
				UNIT – 5			
				CODE OPTIMIZATION			
1	17/4/23	6	43	Principal Sources of Optimization	T1	584	28/4/23
2	19/4/23	5	44	Peep-hole optimization	T1	549	28/4/23
3	24/4/23	6	45	DAG	T1	533	29/4/23
4	26/4/23	2	46	Types of Virtual Machines and their Implementations	T1	567	2/5/23
5	26/4/23	2,5	48	Optimization of Basic Blocks	T1	618	3/5/23
6	29/4/23	5	49	Global Data Flow Analysis	T1	626	3/5/23
7	3/5/23	2,5	51	Efficient Data Flow Algorithm	T1		6/5/23
8	6/5/23	5	52	Question Bank Discussed	Ref Material		9/5/23
9	8/5/23	6	53	Previous Year University Question Paper Discussed	Ref Material		13/5/23
10	10/5/23	2,5	55	Unit wise important problems solved	Ref Material		17/5/23
				LIST OF EXPERIMENTS			
1	7/2/23 14/2/23	1,2	4	Develop a lexical analyzer to recognize a few patterns in C. (Ex. identifiers, constants, comments, operators etc.). Create a symbol table, while recognizing identifiers.	8/3/23 14/3/23		
2	21/2/23 28/2/23	1,2	8	Implement a Lexical Analyzer using Lex Tool	24/3/23		
3	7/3/23 14/3/23	1,2	12	Implement an Arithmetic Calculator using LEX and YACC	28/3/23		
4	21/3/23 25/3/23	1,2 1,2,3,4	18	Generate three address code for a simple program using LEX and YACC.	11/4/23		
5	28/3/23 11/4/23	1,2	22	Implement simple code optimization techniques (Constant folding, Strength reduction andAlgebraic transformation)	18/4/23		
6	18/4/23 25/4/23	1,2	26	Implement back-end of the compiler for which the three address code is given as input and the 8086 assembly language code is produced as output.	21/4/23		
7	2/5/23	1,2	28	CBS:Create a Lexical analyser to take input from text file and count no of characters, no.of lines, words	23/5/23		
8	9/5/23	1,2	30	Model practical	23/5/23		

Internal Assessment and Activities

Unit	Test	Date	Assignment	Date
I	Internal Assessment Test I	17/3/23	24/2/23 Assignment	24/2/23 20/3/23
II				
II	Internal Assessment Test II	18/4/23	3	5/4/23
III			4	15/4/23
IV,V	Internal Assessment Test III	19/5/23	5	10/5/23

	Prepared by	Checked by	Approved by
Signature		G. Hilb	
Name	Ms N. Anu Lavanya	Dr. C.Sujatha	Dr. D. Senthil Kumaran
Designation	Assistant Professor	HoD/CSE	PRINCIPAL
Date	6.2.23	6/2/23	06/2/23
Remarks			

Initials of the Head of the Department with date	Completion of					Completion of Model Exam /Semester
	Unit I	Unit II	Unit III	Unit IV	Unit V	
	4.3.23	20/3/23	15/4/23	28/4/23	17/5/23	
	Internal Test-1		Internal Test-2		Internal Test-3	

Signature with Date	9/6/23	12/6/23	12/6/23
	Faculty-in-charge	Head of the Department	Principal

SSM INSTITUTE OF ENGINEERING AND TECHNOLOGY

Dindigul- Palani Highway, Dindigul – 624 002.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Year/Sem: III/VI

Subject Code/Name: CS8602/Compiler Design

Course Material Link:

https://classroom.google.com/c/NTEzMzgyMjg1NzYy?cjc=tjqt25n

Faculty-Incharge

HoD/CSE

SSM INSTITUTE OF ENGINEERING AND TECHNOLOGY

Dindigul – Palani Highway, Dindigul – 624 002.

ANNA UNIVERSITY EXAMINATION QUESTION PAPER – FEEDBACK FORM

Course Code & Title : CS8602 Compiler design Class III / VI

Faculty Name : N. Anu Lovanya Exam Date : 23/6/23

Q No.	Internal Test	Tutorial	Class Work	Q No.	Internal Test	Tutorial	Class Work
1			✓	11. a			✓
2			✓	11. b	✓		
3	✓			12. a		✓	
4	✓	✓		12. b	✓		
5	✓			13. a	✓		
6		✓		13. b			✓
7	✓			14. a	✓		
8	✓			14. b			✓
9	✓			15. a	✓		
10			✓	15. b	✓		
				16. a			✓
				16.b			✓

Note: Put appropriate Tick (✓) marks in the corresponding box

General Comments by Course In charge:

1. Details of out of syllabus questions (If any): Nil
2. Whether any information needs to be represented to the University on your QP? NO
   If yes, please upload the description of discrepancies in the web portal with Principal Approval.
3. If there is any student absenteeism, give the details and communicate the same to their parents. Nil

Comments by Students :

Student's Name	Feedback / Comments		Expected Marks (out of 100)	Signature
	Part A	Part B & C		
Abinaya. T	Not bad	okay	50	Abi
Preethiga. M.	Average	Average	60	
Kasunya M.D	Average	Average	60	
Evans Abraham. J	Average	Average	60	
Mohamed Hadhi. S	Average	Average	55	

Overall Feedback by Course In charge: Average

Expected Pass Percentage: 75

Course Incharge **HoD** 23/6/23 **Principal**

Note: Please attach University question paper along with all internal test question paper(s)

SSM INSTITUTE OF ENGINEERING AND TECHNOLOGY

Dindigul – Palani Highway, Dindigul – 624 002.

ANNA UNIVERSITY EXAMINATION QUESTION PAPER – FEEDBACK FORM

Course Code & Title : CS8602 Compiler design Class : III / VI

Faculty Name : N. Anu Lavanya Exam Date : 23/6/23

Q No.	Internal Test	Tutorial	Class Work	Q No.	Internal Test	Tutorial	Class Work
1			✓	11. a			✓
2			✓	11. b	✓		
3	✓			12. a		✓	
4	✓	✓		12. b	✓		
5	✓			13. a	✓		
6		✓		13. b			✓
7	✓			14. a	✓		
8	✓			14. b			✓
9	✓			15. a	✓		
10			✓	15. b	✓		
				16. a			✓
				16.b			✓

Note: Put appropriate Tick (✓) marks in the corresponding box

General Comments by Course In charge:

1. Details of out of syllabus questions (If any): Nil
2. Whether any information needs to be represented to the University on your QP? NO
   If yes, please upload the description of discrepancies in the web portal with Principal Approval.
3. If there is any student absenteeism, give the details and communicate the same to their parents. Nil

Comments by Students :

Student's Name	Feedback / Comments		Expected Marks (out of 100)	Signature
	Part A	Part B & C		
Abinaya. T	Not bad	okay	50	Abi
Preethiga. M.	Average	Average	60	
Karunya M.D	Average	Average	60	
Evans Abraham. J	Average	Average	60	
Mohamed Hadhi. S	Average	Average	55	

Overall Feedback by Course In charge: Average

Expected Pass Percentage: 75

Course Incharge **HoD** 23/6/23 **Principal**

Note: Please attach University question paper along with all internal test question paper(s)

Reg. No. : | | | | | | | | | | | | |

Question Paper Code : 50434

B.E./B.Tech. DEGREE EXAMINATIONS, APRIL/MAY 2023.

Sixth Semester

Computer Science and Engineering

CS 8602 — COMPILER DESIGN

(Common to Computer Science and Business Systems)

(Regulations 2017)

Time : Three hours | Maximum : 100 marks

Answer ALL questions.

PART A — (10 × 2 = 20 marks)

1. State the role of lexical analyzer. Identify the lexemes and their corresponding tokens in the following statement: printf ("Total = %d\n", score);

2. What is the difference between compiler and interpreter?

3. For what type of grammar, recursive descent parser cannot be constructed? Show the steps involved in recursive descent parsing with backtracking for the string cad with the given grammar: S -> cAd A -> ab | a?

4. Construct a parse tree and syntax tree for 4–6/3*5+7.

5. What are syntax directed translation schemes?

6. Determine the types and relative addresses for the identifiers in the following sequence of declarations:

```
float x;
record { float x; float y; } p;
record { int tag; float x; float y; } q;
```

7. What is static allocation strategy? State its limitations.

8. State how a task is divided between calling and called program for stack updating?

9. What is peephole optimization?

10. What is a flow graph? State its role in compilation process.

PART B — (5 ×13 = 65 marks)

11. (a) List out the functions of a Lexical Analyzer? State the reasons for the Separation of analyses of programs into Lexical, Syntax, and Semantic Analyses. (13)

Or

(b) Discuss the phases of a compiler indicating the inputs and outputs of each phase in translating the statement "amount = principle + rate * 36.0". (13)

12. (a) Explain the usage of YACC parser generator in construction of a Parser with one example. (13)

Or

(b) Define an LL(1) grammar. How do you check whether the grammar is LL (1) or not? Generate LL(1) parsing table for the Grammar

$S \rightarrow iEtS \mid iEtSeS \mid a$

$E \rightarrow b$

Is Grammar in LL(1) or not? (13)

13. (a) Define syntax tree. What is s-attributed definition? Explain construction of syntax tree for the expression a-4+c using SDD. (13)

Or

(b) With a neat diagram explain the format of the Symbol Table. Discuss the tree structures representation of scope information. (13)

14. (a) Discuss how induction variables can be detected and eliminated from the given intermediate code

B2: i:= i+1

t1: = 4*j

t2: = a[t1]

if t2 < 10 goto B2 (13)

Or

(b) What is an activation record? Explain stack allocation of activation records using example. (13)

15. (a) Explain different code optimization techniques available in local and global optimizations? (13)

Or

(b) Construct the DAG for the following basic block :

(i) t1:=4*i

(ii) t2:=a[t1]

(iii) t3:=4*i

(iv) t4:=b[t3]

(v) t5:=t2*t4

(vi) t6:=prod+t5

(vii) prod:=t6

(viii) t7:=i+1

(ix) i:=t7

(x) if i<=20 goto (i) (13)

PART C — (1 × 15 = 15 marks)

16. (a) Consider the following basic block of 3-address instructions:

a:=b+c x:=a+b b:=a–d c:=b+c d:=a–d y:=a–d

Write the next-use information for each line in the basic block. (15)

Or

(b) Draw transition diagrams corresponding to production rules for arithmetic expressions consisting of operators + and ^ for predictive parser. Explain how parsing takes place for the same using transition diagrams. (15)

———————

8. State how a task is divided between calling and called program for stack updating?

9. What is peephole optimization?

10. What is a flow graph? State its role in compilation process.

PART B — (5 ×13 = 65 marks)

11. (a) List out the functions of a Lexical Analyzer? State the reasons for the Separation of analyses of programs into Lexical, Syntax, and Semantic Analyses. (13)

Or

(b) Discuss the phases of a compiler indicating the inputs and outputs of each phase in translating the statement "amount = principle + rate * 36.0". (13)

12. (a) Explain the usage of YACC parser generator in construction of a Parser with one example. (13)

Or

(b) Define an LL(1) grammar. How do you check whether the grammar is LL (1) or not? Generate LL(1) parsing table for the Grammar

$S \rightarrow iEtS \mid iEtSeS \mid a$

$E \rightarrow b$

Is Grammar in LL(1) or not? (13)

13. (a) Define syntax tree. What is s-attributed definition? Explain construction of syntax tree for the expression a-4+c using SDD. (13)

Or

(b) With a neat diagram explain the format of the Symbol Table. Discuss the tree structures representation of scope information. (13)

14. (a) Discuss how induction variables can be detected and eliminated from the given intermediate code

B2: i:= i+1

t1: = 4*j

t2: = a[t1] .

if t2 < 10 goto B2 (13)

Or

(b) What is an activation record? Explain stack allocation of activation records using example. (13)

15. (a) Explain different code optimization techniques available in local and global optimizations? (13)

Or

(b) Construct the DAG for the following basic block :

(i) t1:=4*i

(ii) t2:=a[t1]

(iii) t3:=4*i

(iv) t4:=b[t3]

(v) t5:=t2*t4

(vi) t6:=prod+t5

(vii) prod:=t6

(viii) t7:=i+1

(ix) i:=t7

(x) if i<=20 goto (i) (13)

PART C — (1 × 15 = 15 marks)

16. (a) Consider the following basic block of 3-address instructions:

a:=b+c x:=a+b b:=a–d c:=b+c d:=a–d y:=a–d

Write the next-use information for each line in the basic block. (15)

Or

(b) Draw transition diagrams corresponding to production rules for arithmetic expressions consisting of operators + and ^ for predictive parser. Explain how parsing takes place for the same using transition diagrams. (15)

———————

b) Consider the code

```
01        a = 1
02        b = 2
03 L0:    c = a + b
04        d = c – a
05        if c < d goto L2
06L1:     d = b + d
07        if d < 1 goto L3
08L2:     b = a + b
09        e = c – a
10        if e = 0 goto L0
11        a = b + d
12        b = a – d
13        goto L4
14L3:     d = a + b
15        e = e + 1
16        goto L3
17L4:     return
```

For the code shown above, determine the following :

i) The basic blocks of instructions (3)

ii) The control-flow graph (CFG) (3)

iii) For each variable, its corresponding def-use chain (3)

iv) The live variables at the end of each basic block. You do not need to determine the live variables before and after each basic block and justify your answer for the value presented for the basic block containing instructions at line 6 and 7. (3)

v) Is the live variable analysis a forward or backward data-flow analysis problem ? Why and what does guarantee its termination when formulated as a data-flow analysis iterative problem ? (3)

15/11/19 /AN

Reg. No. :

Question Paper Code : 91408

B.E./B.Tech. DEGREE EXAMINATIONS, NOVEMBER/DECEMBER 2019

Sixth Semester

Computer Science and Engineering

CS 6660 – COMPILER DESIGN

(Common to Information Technology)

(Regulations 2013)

(Also common to PTCS 6660 – Compiler Design for B.E. (Part-Time) Fifth Semester – Computer Science and Engineering – Regulations 2014)

Time : Three Hours Maximum : 100 Marks

Answer ALL questions

PART – A **(10×2=20 Marks)**

1. Name a compiler construction tool used to design i) Lexical Analyser and ii) Parser.

2. Which phase (or phases) of a compiler
   i) is/are considered the "back end" ?
   ii) access (es) the symbol table (for reading or writing) ?
   iii) check (s) for type mismatches ?
   iv) is/are independent of the underlying machine ?

3. Consider the language of all strings from the alphabet {a, b, c} containing the substring "abcabb". Write a regular expression that describes this language.

4. Give any two reasons for keeping lexical analyser a separate phase instead of making it an integral part of syntax analysis.

5. Show that the grammar, $S \rightarrow aSbS \mid bSaS \mid \varepsilon$ is ambiguous.

6. Give the structure of YACC program to design a simple syntax analyser.

7. Differentiate implicit and explicit type conversions.

8. What is an activation record ? Give the structure of an activation record.

9. Construct a DAG for the following code

   a = b + c

   b = a – d

   c = b + c

   d = a – d

10. What is the cost of the following sequences of instructions ?

    i) MOV b, a
       ADD c, a (1)

    ii) MOV * R1, *R0
        ADD * R2, *R0 (1)

PART – B (5×13=65 Marks)

11. a) What are the different phases of a compiler ? Write their functions. Show how the high level language statement position = initial + rate * 60 is converted to machine code by each phase.

    (OR)

    b) What are the components of a language processing system ? Explain the role of each of these components in a typical compilation and execution of the program.

12. a) Given the regular expression (a | b)*abb over the alphabet Σ = {a, b}

    i) Construct a NFA with ε-transitions using Thompson Construction. (4)

    ii) Convert the NFA obtained from (i) to non-minimal DFA. (5)

    iii) Minimize the number of states obtained from ii) to minimal DFA. (4)

    (OR)

    b) Write a lex program to implement a calculator. Describe the actions of the program and the functions defined and used.

13. a) Consider the following grammar G :

    $S' \rightarrow S$

    $S \rightarrow CC$

    $C \rightarrow cC \mid d$

    Construct the LALR parsing table for the grammar G. Show the moves of the parser on the string ccd

    (OR)

    b) Explain the construction of predictive parsing table and describe the moves of the parser on an input string. Design a predictive parser for the following grammar

    $E \rightarrow E + T \mid T$

    $T \rightarrow T * F \mid F$

    $F \rightarrow (E) \mid id$

    Show the moves of the parser on the input id + id * id.

14. a) What is a syntax tree ? Describe construction of syntax trees for expressions ? Give examples to support your description. Write syntax directed definition for constructing syntax trees/Draw an annotated parse tree for the expression a – 4 + c.

    (OR)

    b) What is a symbol table ? What type of information is stored in it ? Discuss on the use of the data structures i. arrays ii. Linked lists iii. Binary search trees for implementing a symbol table.

15. a) What are the principal sources of optimization ? Explain with suitable examples.

    (OR)

    b) Write a simple code generator algorithm. With an example code, show how the algorithm generates code.

PART – C (1×15=15 Marks)

16. a) Consider the following grammar G :

    $S \rightarrow XaY \mid Y$

    $X \rightarrow bY \mid c$

    $Y \rightarrow X$

    i) Discuss the various steps involved in the construction of SLR parsing. (3)

    ii) Show the canonical collection of LR(0) items. (5)

    iii) Construct the SLR parsing table. (4)

    iv) Show the action of the parser on the input string cac$. (3)

    (OR)

Reg. No. : 9 5 3 4 1 6 1 0 4 0 2 4

Question Paper Code : 52873

B.E./B.Tech. DEGREE EXAMINATIONS, APRIL/MAY 2019.

Sixth Semester

Computer Science and Engineering

CS 6660 — COMPILER DESIGN

(Common to Information Technology)

(Regulation 2013)

(Also common to PTCS 6660 – Complier Design for B.E. (Part-Time) for Fifth Semester – Computer Science and Engineering – Regulation 2014)

Time : Three hours Maximum : 100 marks

Answer ALL questions.

PART A — (10 × 2 = 20 marks)

1. List the attributes stored in symbol table.

2. Why is compiler optimization essential?

3. Discriminate tokens, patterns and lexemes.

4. Write the regular expression for all valid identifiers.

5. What is meant by handle pruning?

6. Mention the purpose of YACC.

7. What are the various ways of passing a parameter to a function?

8. Write the grammar for flow control statement while-do.

9. Define address descriptor.

10. Write the object code sequence for t:=a+b produced by a typical code generator.

PART III (3 × 10 = 30 marks)

11. (a) Explain various phases of compiler in detail. Write the output of each phase of the compiler for the expression c := a + b * 12. (13)

Or

(b) (i) What are the characteristics of compiler construction tools? Explain how compiler construction tools help in implementation of various phases of a compiler. (7)

(ii) Differentiate call-by-value and call-by-reference parameter passing mechanisms with suitable examples. (6)

12. (a) (i) Analyze the role of lexical analyzer with suitable examples. (7)

(ii) Draw and explain the transition diagram that recognizes the lexemes matching the token relop (relational operator). (6)

Or

(b) Write the subset construction algorithm. Using the subset construction algorithm, convert the regular expression (a | b)*abb to DFA. (13)

13. (a) Write the algorithm for construction of LR parsing table for a given grammar. Construct the LR parsing table for the following grammar : (13)

$E \rightarrow E + T$

$E \rightarrow T$

$E \rightarrow T * F$

$T \rightarrow F$

$F \rightarrow (E)$

$F \rightarrow id$

Or

(b) Write the algorithm for construction of LALR parsing table for a given grammar. Using the algorithm for construction of LALR parsing table construct the LALR parsing table for the following grammar. (13)

$S' \rightarrow S$

$S \rightarrow aAd \mid bBd \mid aBe \mid bAe$

$A \rightarrow c$

$B \rightarrow c$

14. (a) (i) Describe syntax-directed translation schemes with appropriate examples. (7)

(ii) Explain how type conversion is performed with suitable examples. (6)

Or

(b) Explain various techniques for storage allocation with examples. (13)

15. (a) (i) Write and explain the algorithm for construction of basic block. (5)

(ii) Construct the DAG for the following basic block. (10)

x = a[i]

a[i] = x

z = a[i]

Or

(b) Explain the algorithm that generates code for a single basic block with suitable examples. (15)

PART C — (1 × 15 = 15 marks)

16. (a) In SQL, keywords and identifiers are case-insensitive. Write a *Lex* program that recognizes the keywords SELECT, FROM, and WHERE (in any combination of capital and lower-case letters), and token ID, which may be any sequence of letters and digits, beginning with a letter. (15)

Or

(b) A simple matrix-multiplication program is given below :

```
for (i=0; i<n; i++)
    for (j=0; j<n; j++)
        c[i][j] = 0.0;
for (i=0; i<n; i++)
    for (j=0; j<n; j++)
        for (k=0; k<n; k++)
            c[i][j] = c[i][j] + a[i][k]*b[k][j];
```

(i) Translate the program into three-address statements. Assume the matrix entries are numbers that require 8 bytes, and that matrices are stored in row-major order. (7)

(ii) Construct the flow graph for the code from 1. (6)

(iii) Identify the loops in the flow graph from 2. (2)

———————

13/12/18

Reg. No. :

Question Paper Code : 20374

B.E./B.Tech. DEGREE EXAMINATION, NOVEMBER/DECEMBER 2018.

Sixth Semester

Computer Science and Engineering

CS 6660 — COMPILER DESIGN

(Common to Information Technology)

(Regulations 2013)

(Also common to PTCS 6660 — Compiler Design – for B.E. (Part-Time) Fifth Semester – Computer Science and Engineering – Regulations 2014)

Time : Three hours Maximum : 100 marks

Answer ALL questions.

PART A — (10 × 2 = 20 marks)

1. Recall the basic the two parts of a compilation process.
2. How a source code is translated to machine code?
3. State the rules to define regular expression.
4. Construct Regular expression for the language $L = \{w \in \{a, b\} / w \text{ ends in } abb\}$.
5. What are the different stages that a parser can recover from a syntactic error?
6. Define LR (0) item.
7. List three kinds of intermediate representation.
8. When procedure call occurs, what are the steps taken?
9. State the problems in code generation.
10. Define common sub expression.

PART B — (5 × 13 = 65 marks)

11. (a) Write short notes about :

(i) Compiler Construction Tools. (7)

(ii) Lexeme, token and pattern. (6)

Or

(b) Discuss in detail about the operations of compiler which transforms the source program from one representation into another. Illustrate the output for the input : (13)

$a = (b + c) * (b + c) * 2$.

12. (a) Write briefly about :

(i) the role of Lexical analyzer with the possible error Recovery actions. (5)

(ii) recognition and specification of tokens. (8)

Or

(b) Construct the minimized DFA for the regular expression $(0+1)*(0+1)01$. (13)

13. (a) Show that the following grammar

$S \rightarrow Aa \mid bAc \mid dc \mid bda$

$A \rightarrow a$

is LALR(1) but not SLR(1). (13)

Or

(b) Show that the following grammar

$S \rightarrow Aa \mid bAc \mid Bc \mid bBa$

$A \rightarrow d$

$B \rightarrow d$

is LR(1) but not LALR(1). (13)

14. (a) Apply the S-attributed definition and constructs syntax trees for a simple expression grammar involving only the binary operators + and −. As usual, these operators are at the same precedence level and are jointly left associative. All nonterminal have one synthesized attribute node, which represents a node of the syntax tree.

Production: $E \rightarrow E_1 + T$, $E \rightarrow T$, $T \rightarrow (E)$, $T \rightarrow$ id/num . (13)

Or

(b) Discuss in detail about :

(i) Storage allocation strategies. (7)

(ii) Parameter passing methods. (6)

15. (a) Discuss in detail about optimization of basic blocks. (13)

Or

(b) Explain in detail about issues in the design of a code generator. (13)

PART C — (1 × 15 = 15 marks)

16. (a) Suppose we have a production $A \rightarrow B\ C\ D$. Each of the four nonterminals has two attributes s, which is synthesized, and i, which is inherited. For each set of rules below, check whether the rules are consistent with (i) an S-attributed definition, (ii) an L-attributed definition (iii) any evaluation order at all.

(1) $A.s = B.i + C.i$

(2) $A.s = B.i + C.s$ and $D.i = A.i + B.s$

(3) $A.s = B.s + D.s$

(4) $A.s = D.i$

$B.i = A.s + C.s$

$C.i = B.s$

$D.i = B.i + C.i$. (15)

Or

(b) Construct a Syntax-Directed Translation scheme that translates arithmetic expression from infix into postfix notation. (15)

04/11/17 – (AN)

Reg. No. : | | | | | | | | | | | | |

Question Paper Code : 50398

B.E./B.Tech. DEGREE EXAMINATION, NOVEMBER/DECEMBER 2017

Sixth Semester

Computer Science and Engineering

CS6660 – COMPILER DESIGN

(Common to : Information Technology)

(Regulations 2013)

Time : Three Hours | Maximum : 100 Marks

Answer ALL questions

PART – A (10×2=20 Marks)

1. What is an interpreter ?
2. What do you mean by Cross-Compiler ?
3. What is the role of lexical analysis phase ?
4. Define Lexeme.
5. Draw syntax tree for the expression a=b*– c+b*– c.
6. What are the three storage allocation strategies ?
7. Differentiate NFA and DFA.
8. Compare syntax tree and parse tree.
9. Draw the DAG for the statement a = (a*b+c)–(a*b+c).
10. What are the properties of optimizing compilers ?

PART – B (5×16=80 Marks)

11. a) What are compiler construction tools ? Write note on each Compiler Construction tool.

(OR)

b) Explain in detail the various phases of compilers with an example.

PART – B (5 × 16 = 80 Marks)

11. (a) Describe the various phases of compiler and trace it with the program segment (position:= initial + rate * 60). **(16)**

OR

(b) (i) Explain language processing system with neat diagram. **(8)**
(ii) Explain the need for grouping of phases. **(4)**
(iii) Explain various Error encountered in different phases of compiler. **(4)**

12. (a) (i) Differentiate between lexeme, token and pattern. **(6)**
(ii) What are the issues in lexical analysis ? **(4)**
(iii) Write notes on regular expressions. **(6)**

OR

(b) (i) Write notes on regular expression to NFA. Construct Regular expression to NFA for the sentence (a\|b)* a. **(10)**
(ii) Construct DFA to recognize the language (a/b)* ab. **(6)**

13. (a) (i) Construct Sack implementation of shift reduce parsing for the grammar **(8)**
E -> E+E
E -> E*E
E -> (E)
E -> id and the input string id1 + id2 *id3
(ii) Explain LL(1) grammar for the sentence S->iEts \| iEtSeS \| a E->b. **(8)**

OR

(b) (i) Write an algorithm for Non recursive predictive parsing. **(6)**
(ii) Explain Context free grammers with examples. **(10)**

14. (a) (i) Construct a syntax directed definition for constructing a syntax tree for assignment statements. **(8)**
S → **id** : = E
E → E1 + E2
E →E1 * E2
E →-E1
E →(E1)
E →**id**

(ii) Discuss specification of a simple type checker. **(8)**

OR

(b) Discuss different storage allocation strategies. **(16)**

15. (a) Explain Principal sources of optimization with examples. **(16)**

OR

(b) (i) Explain various issues in the design of code generator. **(8)**
(ii) Write note on simple code generator. **(8)**

Reg. No. : | | | | | | | | | | | | |

Question Paper Code : 90421

B.E./B.Tech. DEGREE EXAMINATIONS, NOVEMBER/DECEMBER 2022.

Sixth Semester

Computer Science and Engineering

CS 8602 — COMPILER DESIGN

(Regulations 2017)

Time : Three hours Maximum : 100 marks

Answer ALL questions.

PART A — (10 × 2 = 20 marks)

1. Write a regular expression to represent all possible numbers (integer, float, and exponential)

2. How and why input buffering is occurring?

3. Write down the context-free grammar for representing the if-else statement of any language.

4. Write LMD for the word $(i+i)/(i*i)$ using the grammar, $E \rightarrow E+E \mid E/E \mid E*E \mid (E) \mid i$

5. What is the use of a Three Address Code?

6. What is type-checking?

7. How is the liveliness of a variable calculated?

8. List the different types of storage allocation strategies.

9. Define constant folding.

10. What is the difference between peephole optimization and normal optimization?

PART B — (5 × 13 = 65 marks)

11. (a) An online shopping site has set up the following criteria for setting the password for viewing their products

(i) Password should be 3 to 5 characters long using alphabets and numerals.

(ii) It should start with alphabet

(iii) Two special symbols (% and #) are permitted

(iv) At least one capital letter should be present

Assume the password given by the user has to be scanned by the first phase of the compiler. Represent the above in an appropriate format and convert it into a finite state machine.

Or

(b) What are the five token formats of a programming language? Represent each token format using regular expression and draw equivalent finite automata.

12. (a) Construct LR (0) items for the following grammar, G :

S→S+R|R
R→R*T|T
T→(S)|i

Or

(b) Construct LL(1) parsing table for the following grammar, G and check whether the word "ia<btd" is a string of the grammar or not

S→iEtS | iEtSeS | d
E→id OP id
OP→<|>|< = |>=| = = |!=

13. (a) Write the semantic actions for the declaration statements in the following piece of code. Show the symbol tables created.

```
Procedure a ( )
{
  int a, b;
 float c;
 procedure b ( )
  {
  int a ;
  float c;
  procedure c ( )
 {
    int d;
  }
 }
}
```

Or

(b) Translate the following piece of code into Three Address Codes using syntax-directed translation.

```
Count : = 0;
read x;
if (x > 0 & & x < 10)
then
        count : = count + 1;
        read x;
else
        count : = 0;
endif.
```

14. (a) (i) Elaborate on the issues in the design of code generator. (7)

(ii) Write the algorithm used in code generation phase of compiler. (6)

Or

(b) Elaborate on the storage allocation strategies.

15. (a) What are the principal Sources of Optimization? Explain the local optimization strategies with appropriate examples.

Or

(b) What is a peephole? Explain the optimizations that can be performed on a peephole

PART C — (1 × 15 = 15 marks)

16. (a) For calculating the income tax, the following formulae is used by a concern.

Tax = (basic_pay+DA+HRA) * 0.3

Where,

basic_pay is an integer value and DA and HRA could be either integer or floating point numbers.

Elaborate on how this statement is converted into a machine language format while passing through the six phases of the compiler Elaborate on the process by giving the output. Write the use of symbol table an error handling phase too.

Or

(b) For calculating the income tax, the following formulate is used by a concern.

if salary > 500000 then

Tax = salary * 0.20

Else

Tax = salary * 0.10

End if

where,

salary could be either integer or floating point number.

Elaborate on how this statement is converted into a machine language format while passing through the six phases of the compiler. Elaborate on the process by giving the output. Write the use of symbol table an error handling phase too.

Reg. No. : | | | | | | | | | | | | |

Question Paper Code : 20417

B.E./B.Tech. DEGREE EXAMINATIONS, APRIL/MAY 2022.

Fourth/Fifth Semester

Computer Science and Engineering

CS 8493 – OPERATING SYSTEMS

(Common to : Electronics and Communication Engineering/Computer Science and Business Systems/Information Technology)

(Regulations 2017)

Time : Three hours | Maximum : 100 marks

Answer ALL questions.

PART A — (10 × 2 = 20 marks)

1. What are the steps in Instruction Execution?
2. Define System Boot.
3. What are Kernel Threads?
4. Describe Context Switch.
5. What is External Fragmentation?
6. Describe Demand Paging System.
7. List out major attributes of a file.
8. What is Mount point'?
9. Define Kernel in Linux Operating System.
10. What is the purpose of fork() and exec() system calls?

PART B — (5 × 13 = 65 marks)

11. (a) (i) Illustrate the flow of control with and without Interrupts. (7)

(ii) List and explain five types of System Calls. (6)

Or

(b) (i) Explain the basic structure and operations of operating system. (7)

(ii) Brief about the various types of memories in memory hierarchy. (6)

12. (a) (i) With suitable example, explain about various types of process scheduling algorithms. (7)

(ii) Discuss about the various methods for handling deadlock. (6)

Or

(b) (i) Describe Critical Section Problem with a suitable example. (7)

(ii) Explain with a neat diagram about various multi threading models. (6)

13. (a) Explain briefly about the hardware implementation of Page Table.

Or

(b) Consider the page reference string 7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2, 0, 1, 7, 0, 1. Calculate the number of page faults could occur for Optimal Page replacement algorithm and LRU page replacement algorithm.

14. (a) Describe about the Sequential and Direct Access methods of information from a file.

Or

(b) How disc space is allocated in Contagious Allocation method? What is the drawback of this method?

15. (a) Illustrate the various components that make up a Full Linux System with a neat diagram.

Or

(b) Explain in detail about the Android Architecture and its components.

PART C — (1 × 15 = 15 marks)

16. (a) Consider the following status of the system.

	Allocation				Max				Available			
	A	B	C	D	A	B	C	D	A	B	C	D
P0	0	0	1	2	0	0	1	2	1	5	2	0
P1	1	0	0	0	1	7	5	0				
P2	1	3	5	4	2	3	5	6				
P3	0	6	3	2	0	6	5	2				
P4	0	0	1	4	0	6	5	6				

Answer the following questions using the banker's algorithm:

(i) What is the content of the matrix *Need*?

(ii) Is the system in a safe state?

(iii) If a request from process P1 arrives for (0,4,2,0), can the request be granted immediately?

Or

13. (a) Write the syntax directed translation for the following code:

$E \mapsto E_1$ or E_2

$E \mapsto E_1$ and E_2

$E \mapsto$ not E_1

$E \mapsto (E_1)$

$E \mapsto id_1$ relop id_2

$E \mapsto$ true

$E \mapsto$ false

Or

(b) Write the syntax directed translation for the following piece of code.

while $a < b$

do

if c < d

then

x : = y+z

else

x :=y–z

14. (a) Elaborate the issues in developing a code generator.

Or

(b) Construct the basic blocks and flowgraph for the following piece of code.

```
for i from 1 to 10 do
    for j from 1 to 10 do
    a[i, j] = 0.0;
for i from 1 to 10 do
    a[i,i]=1.0;
```

15. (a) Describe the parameter passing techniques with an example.

Or

(b) Explain the storage allocation techniques with an example.

PART C — (1 × 15 = 15 marks)

16. (a) Consider the following basic block, in which all variables are integers and ** denotes exponentiation

```
a := x**2
b := 3
c := x
d := c * c
e := b * 2
f := a + d
g := e * f
```

Apply the following optimization techniques to this basic block, in order. Compute the result of each transformation.

(i) Algebraic Simplification (3)

(ii) Copy Propagation (3)

(iii) Constant Folding (3)

(iv) Dead Code Elimination (3)

(v) Common Sub-expression Elimination (3)

Or

(b) Construct LR(1) items for the following grammar and draw the transition diagram representing the transitions among CLR items.

$S \rightarrow CC$

$C \rightarrow cC$

$C \rightarrow d$

Show whether the string "cdcd" is accepted by this grammar or not.

ANNA UNIVERSITY :: CHENNAI - 600 025
OFFICE OF THE CONTROLLER OF EXAMINATIONS
Assessment Details Entered - Report : By subject
APRIL / MAY EXAMINATION,2023 - EXAMINATIONS

Inst Code & Name : 9221 - SSM INSTITUTE OF ENGINEERING AND TECHNOLOGY

Subject Code / Name : CS8602 : Compiler Design
University : AUC
Semester : 06

Register No.	Name of the Student	Attend Period1	Total Period1	Attend Period2	Total Period2	IM 2	Attend Period3	Tot Period3	IM 3	Attend Period4	Total Period4	IM 4
922120104001	ABINAYA G	13	16	24	24	86	14	16	76	23	26	92
922120104002	ABINAYA T	16	16	22	24	86	16	16	72	21	26	72
922120104003	AKALYA A N	16	16	22	24	96	14	16	90	26	26	92
922120104004	ANU P	16	16	20	24	94	14	16	86	22	26	90
922120104005	ASHOK KUMAR M	13	16	21	24	64	16	16	72	25	26	75
922120104006	ASMETAA G Y	15	16	22	24	96	12	16	74	26	26	64
922120104007	BALASURYA R	16	16	24	24	70	14	16	72	25	26	76
922120104008	BHOOMIKA R	13	16	24	24	94	14	16	100	22	26	72
922120104009	DINESH RAJA E	14	16	24	24	64	14	16	72	25	26	77
922120104010	EVANS ABRAHAM J	16	16	22	24	72	14	16	70	20	26	76
922120104011	HAREESWARAN S	13	16	24	24	60	14	16	70	21	26	80
922120104012	HARINI S	16	16	22	24	78	14	16	72	25	26	64
922120104013	JEEVA J	16	16	22	24	78	16	16	76	26	26	63
922120104014	JEYASHREE S	14	16	22	24	70	16	16	74	23	26	74
922120104015	JOHANS PRAVEEN S	16	16	21	24	70	16	16	76	22	26	76
922120104016	JOSEPHINE JESILA M	14	16	24	24	90	16	16	70	24	26	63
922120104018	KARUNYA M D	14	16	24	24	84	14	16	74	26	26	65
922120104019	KARUPPAIYA M	16	16	24	24	64	14	16	74	25	26	75
922120104022	LOGATHARANI S	15	16	17	24	86	12	16	78	26	26	74
922120104023	LOGESHWARI P	15	16	22	24	96	14	16	76	26	26	84
922120104024	LOKESH G	16	16	19	24	76	16	16	92	26	26	90
922120104025	MADHESH KUMAR D	13	16	21	24	78	16	16	78	24	26	65
922120104026	MOHAMED ARSATH M	11	16	18	24	70	14	16	78	19	26	62
922120104027	MOHAMED FAZIL J	13	16	21	24	60	12	16	68	21	26	83
922120104028	MOHAMED HADHI S	16	16	24	24	64	14	16	90	19	26	86
922120104029	MUGILAN M	11	16	20	24	62	16	16	72	23	26	78
922120104030	PARTHI PRASATH N	15	16	21	24	64	14	16	80	25	26	79
922120104031	PIRUTHVI RAMANA V	11	16	19	24	70	16	16	90	25	26	60
922120104032	POOJA M	15	16	23	24	96	16	16	100	26	26	100
922120104033	PRADEEP V	15	16	22	24	96	16	16	98	24	26	94
922120104034	PRAKASH S	15	16	22	24	66	14	16	68	21	26	79
922120104035	PREETHIGA M	16	16	22	24	90	14	16	80	24	26	66
922120104036	PRETHEEBA U	15	16	24	24	96	16	16	100	25	26	100
922120104037	REENA M	13	16	20	24	94	16	16	84	22	26	62
922120104038	SABARIKRISHNAN R	10	16	20	24	74	16	16	74	22	26	88
922120104039	SAKTHI VIGNESHWARAN B	16	16	24	24	96	16	16	84	24	26	80
922120104040	SANJAY PANDI M	12	16	18	24	68	16	16	78	23	26	75
922120104041	SANJEEV SARAVANAN S	11	16	19	24	60	16	16	70	23	26	82
922120104042	SANTHIYADHARSHINI S	16	16	19	24	92	16	16	70	26	26	92
922120104043	SANTHOSH R	15	16	19	24	80	16	16	70	26	26	84
922120104044	SARAN PANDIAN S	14	16	19	24	66	14	16	70	22	26	78
922120104045	SATHEESH KUMAR K	16	16	24	24	76	16	16	70	24	26	70
922120104046	SHALINI J	16	16	22	24	94	16	16	92	26	26	75
922120104047	SHARMILA S	16	16	21	24	96	14	16	100	26	26	96
922120104049	SINDHUJA INFANT A	14	16	21	24	86	14	16	70	22	26	100
922120104050	SIVA SHANTHANA BHARATHI M	16	16	24	24	70	14	16	74	24	26	66
922120104051	SIVASUNDAR V	12	16	19	24	70	16	16	70	19	26	75
922120104052	SOUNDHARYA DEVI M	16	16	21	24	94	16	16	100	26	26	96
922120104053	SRIDHARAN S	12	16	24	24	60	14	16	70	25	26	77
922120104054	SRIRAM J	11	16	23	24	74	12	16	70	26	26	69
922120104055	SRIRAM S	16	16	24	24	82	14	16	74	24	26	65
922120104056	SRIRAM PRASATH L	15	16	16	24	66	14	16	70	26	26	75

ANNA UNIVERSITY :: CHENNAI - 600 025
OFFICE OF THE CONTROLLER OF EXAMINATIONS
Assessment Details Entered - Report : By subject
APRIL / MAY EXAMINATION,2023 - EXAMINATIONS

Inst Code & Name : 9221 - SSM INSTITUTE OF ENGINEERING AND TECHNOLOGY

922120104057	SUBBIRAMANI R	10	16	21	24	64	16	16	70	24	26	76
922120104058	SUBHA S	14	16	23	24	96	16	16	100	26	26	98
922120104059	TAMIL ARASAN K	16	16	24	24	80	16	16	82	26	26	62
922120104060	VARSHINI U	16	16	21	24	96	16	16	100	26	26	100
922120104061	VINOTH KUMAR A	14	16	21	24	72	12	16	74	22	26	76
922120104301	SRIRAM V M	14	16	18	24	60	14	16	70	25	26	82

Internal Marks Report

College Code / Name : 9221 - SSM INSTITUTE OF ENGINEERING AND TECHNOLOGY

Branch Code / Name : 104 - B.E. Computer Science and Engineering

Semester : 06 University : AUC Regulation : 2017

S.No	Register Number	Name	CS8601	CS8602	CS8603	CS8611	CS8651	CS8661	CS8662	CS8691	HS8581	SB8033	SB8040
1	922120104001	ABINAYA G	15	17	15	19	15	18	19	17	16		
2	922120104002	ABINAYA T	14	15	15	19	15	20	20	17	16		
3	922120104003	AKALYA A N	17	19	16	20	15	19	19	19	20		
4	922120104004	ANU P	15	18	15	19	18	19	19	19	18		
5	922120104005	ASHOK KUMAR M	14	14	15	19	14	19	19	15	20		
6	922120104006	ASMETAA G Y	16	16	16	17	15	19	19	15	20		
7	922120104007	BALASURYA R	14	15	15	18	15	18	18	15	20		
8	922120104008	BHOOMIKA R	15	18	15	20	16	19	19	15	20		
9	922120104009	DINESH RAJA E	15	14	15	18	16	19	19	15	18		
10	922120104010	EVANS ABRAHAM J	14	15	15	19	15	19	19	15	16		
11	922120104011	HAREESWARAN S	14	14	15	18	14	19	19	16	20		
12	922120104012	HARINI S	14	14	15	19	14	20	19	17	18		
13	922120104013	JEEVA J	14	14	15	18	15	19	19	15	20		
14	922120104014	JEYASHREE S	15	15	15	18	14	19	19	16	20		
15	922120104015	JOHANS PRAVEEN S	16	15	15	19	17	19	19	17	18		
16	922120104016	JOSEPHINE JESILA M	15	15	16	19	16	20	19	15	20		
17	922120104018	KARUNYA M D	16	15	16	19	16	19	19	15	18		
18	922120104019	KARUPPAIYA M	15	14	15	18	15	18	18	15	20		
19	922120104022	LOGATHARANI S	14	16	15	18	14	19	19	16	20		
20	922120104023	LOGESHWARI P	16	17	15	18	15	19	19	17	16		
21	922120104024	LOKESH G	15	17	15	19	16	20	20	17	16		
22	922120104025	MADHESH KUMAR D	16	15	15	18	15	19	19	15	15		
23	922120104026	MOHAMED ARSATH M	14	14	15	18	14	18	18	15	20		
24	922120104027	MOHAMED FAZIL J	14	14	15	20	14	18	18	15	20		
25	922120104028	MOHAMED HADHI S	15	16	15	18	14	19	19	15	18		
26	922120104029	MUGILAN M	15	14	15	18	16	19	19	15	16		
27	922120104030	PARTHI PRASATH N	15	15	15	19	14	19	19	15	18		
28	922120104031	PIRUTHVI RAMANA V	14	15	15	19	15	19	19	15	20		
29	922120104032	POOJA M	19	20	18	20	19	20	20	20	20		
30	922120104033	PRADEEP V	16	19	17	20	17	20	20	19	20		

Internal Marks Report

College Code / Name : 9221 - SSM INSTITUTE OF ENGINEERING AND TECHNOLOGY

Branch Code / Name : 104 - B.E. Computer Science and Engineering

Semester : 06 University : AUC Regulation : 2017

S.No	Register Number	Name	CS8601	CS8602	CS8603	CS8611	CS8651	CS8661	CS8662	CS8691	HS8581	SB8033	SB8040
31	922120104034	PRAKASH S	15	14	15	19	15	20	19	15	18		
32	922120104035	PREETHIGA M	16	16	15	19	14	20	19	16	20		
33	922120104036	PRETHEEBA U	18	20	16	20	17	20	20	20	20		
34	922120104037	REENA M	14	16	15	18	14	19	19	15	15		
35	922120104038	SABARIKRISHNAN R	16	16	16	19	15	19	19	18	16		
36	922120104039	SAKTHI VIGNESHWARAN B	15	17	15	19	14	20	20	17	18		
37	922120104040	SANJAY PANDI M	15	15	15	18	15	19	19	15	16		
38	922120104041	SANJEEV SARAVANAN S	15	14	15	18	15	19	19	15	18		
39	922120104042	SANTHIYADHARSHINI S	15	17	15	18	17	20	20	16	20		
40	922120104043	SANTHOSH R	15	16	15	20	16	20	20	15	20		
41	922120104044	SARAN PANDIAN S	15	14	15	18	15	19	19	15	16		
42	922120104045	SATHEESH KUMAR K	16	14	15	18	14	20	20	15	16		
43	922120104046	SHALINI J	17	17	16	19	16	20	20	19	20		
44	922120104047	SHARMILA S	18	19	17	20	19	20	20	19	20		
45	922120104049	SINDHUJA INFANT A	15	17	15	19	16	19	19	17	20		
46	922120104050	SIVA SHANTHANA BHARATHI M	15	14	15	19	15	18	18	15	16		
47	922120104051	SIVASUNDAR V	14	14	15	18	14	19	19	15	15		
48	922120104052	SOUNDHARYA DEVI M	17	19	16	19	17	20	20	18	20		
49	922120104053	SRIDHARAN S	15	14	15	18	14	18	18	15	20		
50	922120104054	SRIRAM J	14	14	15	18	15	19	19	16	16		
51	922120104055	SRIRAM S	16	15	15	18	15	19	19	15	18		
52	922120104056	SRIRAM PRASATH L	15	14	15	18	16	19	19	16	18		
53	922120104057	SUBBIRAMANI R	16	14	15	18	15	20	20	15	18		
54	922120104058	SUBHA S	19	20	17	20	19	20	20	20	18		
55	922120104059	TAMIL ARASAN K	15	15	15	19	15	20	20	16	20		
56	922120104060	VARSHINI U	18	20	18	20	19	20	20	19	20		
57	922120104061	VINOTH KUMAR A	15	15	15	17	15	19	19	16	18		
58	922120104301	SRIRAM V M	14	14	15	19	16	19	19	15	18		