Sri Krishna Institute of Technology, Bangalore

COURSE PLAN

Academic Year 2019-2020

Program:	B E – Computer Science & Engineering
Semester :	6
Course Code:	17CS63
Course Title:	System Software and Compiler Design
Credit / L-T-P:	4 / 4-0-0
Total Contact Hours:	50
Course Plan Author:	Geetha Megharaj

Academic Evaluation and Monitoring Cell

Sri Krishna Institute of Technology
#29,Chimney hills,Hesaraghata Main road, Chikkabanavara Post
Bangalore – 560090, Karnataka, INDIA
Phone / Fax :08023721477/28392221/23721315
Web: www.skit.org.in , e-mail: skitprinci@gmail.com

Table of Contents

A. COURSE INFORMATION

1. Course Overview

Degree:	BE	Program:	CS
Semester:	6	Academic Year:	2019-2020
Course Title:	SYSTEM SOFTWARE AND COMPILER DESIGN	Course Code:	17CS63
Credit / L-T-P:	4 / 4-0-0	SEE Duration:	180 Minutes
Total Contact Hours:	50 Hours	SEE Marks:	60 Marks
CIA Marks:	20 Marks	Assignment	1 / Module
Course Plan Author:	Geetha Megharaj	Sign ..	Dt:
Checked By:		Sign ..	Dt:
CO Targets	CIA Target :	SEE Target:	

Note: Define CIA and SEE % targets based on previous performance.

2. Course Content

Content / Syllabus of the course as prescribed by University or designed by institute.

Module	Content	Teaching Hours	Blooms Learning Levels
1	Introduction to System Software, Machine Architecture of SIC and SIC/XE. **Assemblers:** Basic assembler functions, machine dependent assembler features, machine independent assembler features, assembler design options. **Macroprocessors:** Basic macro processor functions, machine independent macro processor features, Macro processor design options, implementation examples	10	-Understand L2 -Apply L3
2	**Loaders and Linkers:** Basic Loader Functions**,** Machine Dependent Loader Features, Machine Independent Loader Features, Loader Design Options, Implementation Examples. **System File and Library Structure**: Introduction, Library And File Organization, Design Of A Record Source Program File Structure, Object Code, Object File, Object File Structure, Executable File, Executable File Structure,Libraries, Image File Structure .	10	-Understand L2 -Apply L3
3	**Lexical Analysis**: Introduction, Alphabets And Tokens In Computer Languages, Representation, Token Recognition And Finite Automata, Implementation, Error Recovery.	10	-Understand L2 -Apply L3
4	**Syntax Analysis:** Introduction, Role Of Parsers, Context Free Grammars, Top Down Parsers, Bottom-Up Parsers, Operator-Precedence Parsing	10	-Understand L2 -Apply L3
5	Syntax Directed Translation, Intermediate code generation, Code generation	10	-Understand L2 -Apply L3
-	**Total**	**50**	**-**

3. Course Material

Books & other material as recommended by university (A, B) and additional resources used by course teacher (C).

1. Understanding: Concept simulation / video ; one per concept ; to understand the concepts ; 15 – 30 minutes
2. Design: Simulation and design tools used – software tools used ; Free / open source

3. Research: Recent developments on the concepts – publications in journals; conferences etc.

Modules	Details	Chapters in book	Availability
A	**Text books (Title, Authors, Edition, Publisher, Year.)**	**-**	**-**
1, 2	System Software by Leland. L. Beck, D Manjula, 3rd edition, 2012	1,2,3, 4	In Lib / In Dept
3,4,5	Compilers-Principles, Techniques and Tools by Alfred V Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman. Pearson, 2nd edition, 2007	1,2,3,4,5, 6,7,8,9	In Lib/ In dept
B	**Reference books (Title, Authors, Edition, Publisher, Year.)**	**-**	**-**
1, 2	Systems programming – Srimanta Pal , Oxford university press, 2016	1,2,3	In Lib
1, 2,3,4	System programming and Compiler Design, K C Louden, Cengage Learning	1,2,3,4,5	Not Available
1,2	System software and operating system by D. M. Dhamdhere TMG.	1,2,3	In lib
3,4,5	Compiler Design, K Muneeswaran, Oxford University Press 2013.	1,2,3,4,5	
C	**Concept Videos or Simulation for Understanding**	**-**	**-**
C1	*https://www.youtube.com/watch?v=VG9VopzV_To*		
C2	Loaders and Linker https://www.youtube.com/watch?v=VG9VopzV_To		
C3	Phases of compiler design https://www.youtube.com/watch?v=Qkwj65l_96I		
C4	Design of Lexical Analyser https://www.youtube.com/watch?v=WccZQSERfCM		
	Top down Parser Design https://www.youtube.com/watch?v=3_VCoBfrt9c		
	Bottom up parser design https://www.youtube.com/watch?v=MWX0-_mHYcc		
	Semantic analysis and intermediate code generation https://www.youtube.com/watch?v=LF6L6pbWJ5U		
D	**Software Tools for Design**	**-**	**-**
E	**Recent Developments for Research**	**-**	**-**
F	**Others (Web, Video, Simulation, Notes etc.)**	**-**	**-**
1	Overview of compiler design https://www.tutorialspoint.com/compiler_design/compiler_design_overview.htm		

4. Course Prerequisites

Refer to GL01. If prerequisites are not taught earlier, GAP in curriculum needs to be addressed. Include in Remarks and implement in B.5.
Students must have learnt the following Courses / Topics with described Content . . .

Modules	Course Code	Course Name	Topic / Description	Sem	Remarks	Blooms Level
3	17CS44	Microprocessor and Microcontroller	Assembly Level Programming	4	-	Understand L2 Apply L3

5. Content for Placement, Profession, HE and GATE

The content is not included in this course, but required to meet industry & profession requirements and help students for Placement, GATE, Higher Education, Entrepreneurship, etc. Identifying Area / Content requires experts consultation in the area.
Topics included are like, a. Advanced Topics, b. Recent Developments, c. Certificate Courses, d. Course Projects, e. New Software Tools, f. GATE Topics, g. NPTEL Videos, h. Swayam videos etc.

Modules	Topic / Description	Area	Remarks	Blooms Level

1	Assembler Simulation	Placement, Entrepreneurship		Understand L2 Apply L3
2	Loader simulation	Placement, Entrepreneurship		Understand L2 Apply L3
3	Lexical Analyser Implementation	Placement, Entrepreneurship		Understand L2 Apply L3
4	Parser Implementation	Placement, GATE, Higher Study, Entrepreneurship		Understand L2 Apply L3

B. OBE PARAMETERS

1. Course Outcomes

Expected learning outcomes of the course, which will be mapped to POs.

Modules	Course Code.#	Course Outcome **At the end of the course, student should be able to . . .**	Teach. Hours	Instr Method	Assessment Method	Blooms' Level
1	CO1	Illustrate working of Assemblers, Macroprocessors	10	Lecture PPT	Assignment Unit Test	Understand Apply L2,L3
2	CO2	Illustrate working of Loader and Linker	10	Lecture PPT	Assignment Unit Test	Understand Apply L2,L3
3	CO3	Explain Lexical analyzers	10	Lecture PPT	Assignment Unit Test	Understand Apply L2,L3
4	CO4	Design Parsers	10	Lecture PPT	Assignment Unit Test	Understand Apply L2,L3
5	CO5	Discuss Syntax Directed Translation and Code Generators	10	Lecture PPT	Assignment Unit Test	Understand Apply L2,L3
-	**-**	**Total**	**50**	**-**	**-**	**L2-L3**

2. Course Applications

Write 1 or 2 applications per CO.
Students should be able to employ / apply the course learnings to . . .

Modules	Application Area Compiled from Module Applications.	CO	Level
1	Design Macro Processor	CO1	L3
1	Design Assemblers	CO1	L3
2	Design Loaders and Linkers	CO2	L3
3	Design and develop Lexical Analysis Tools	CO3	L3
4	Design Parsers	CO4	L3

3. Articulation Matrix

CO – PO Mapping with mapping level for each CO-PO pair, with course average attainment.

-	-	Course Outcomes	Program Outcomes														-
Modules	CO.#	**At the end of the course student should be able to . . .**	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	Level
1	CO1	Illustrate working of Assemblers, Macroprocessors	3	3	3	2					3		2	3	1	3	2	
2	CO2	Illustrate working of Loader and Linker	3	2	2								2	3		2		
3	CO3	Explain Lexical analyzers	3	3	2	1					3		2	3		3		

4	CO4	Design Parsers	3	3	2	1	3				3		2	3		3	3	
5	CO5	Discuss Syntax Directed Translation and Code Generators	3	2	1									3		2		
-	**17CS63**	Average																-
-	*PO, PSO*	*1.Engineering Knowledge; 2.Problem Analysis; 3.Design / Development of Solutions; 4.Conduct Investigations of Complex Problems; 5.Modern Tool Usage; 6.The Engineer and Society; 7.Environment and Sustainability; 8.Ethics; 9.Individual and Teamwork; 10.Communication; 11.Project Management and Finance; 12.Life-long Learning; S1.Develop Computer Applications ; S2. Entrepreneur S3. Use of Open Ended Programming Environment*																

4. Curricular Gap and Content

Topics & contents not covered (from A.4), but essential for the course to address POs and PSOs.

Modules	Gap Topic	Actions Planned	Schedule Planned	Resources Person	PO Mapping
1					
2					

C. COURSE ASSESSMENT

1. Course Coverage

Assessment of learning outcomes for Internal and end semester evaluation.

Modules	Title	Teach. Hours	No. of question in Exam						CO	Levels
			CIA-1	CIA-2	CIA-3	Asg	Extra Asg	SEE		
1	Introduction to System Software, Machine Architecture of SIC and SIC/XE. Assemblers: Macroprocessors:	10	2	-	-	1	1	2	CO1,CO2	L2,L3
2	Loaders and Linkers: System File and Library Structure:	10	2	-	-	1	1	2	CO3	L2,L3
3	Lexical Analysis:	10	-	2	-	1	1	2	CO4	L2,L3
4	Syntax Analysis:	10	-	2	-	1	1	2	CO4	L2,L3
5	Syntax Directed Translation, Intermediate code generation, Code generation	10	-	-	4	1	1	2	CO5	L2,L3
-	**Total**	**50**	**4**	**4**	**4**	**5**	**5**	**10**	-	-

2. Continuous Internal Assessment (CIA)

Assessment of learning outcomes for Internal exams. Blooms Level in last column shall match with A.2.

Modules	Evaluation	Weightage in Marks	CO	Levels
1, 2	CIA Exam – 1	30	CO1, CO2	L2,L3
3, 4	CIA Exam – 2	30	CO3,CO4	L2,L3
5	CIA Exam – 3	30	CO5	L2,L3
1	Assignment - 1	10	CO1, CO2	L2,L3
2	Assignment - 2	10	CO3,CO4	L2,L3
3	Assignment - 3	10	CO5	L2,L3
	Final CIA Marks	**40**	-	-

D1. TEACHING PLAN - 1

Module - 1

Title:	**Introduction to System Software**	Appr Time:	10 Hrs
a	***Course Outcomes*** *The student should be able to:*	**CO**	**Blooms**
1	Design Assemblers	CO1	**L2,L3**
2	Illustrate working of Macroprocessors	CO1	L2,L3
b	***Course Schedule***	-	-
Class No	**Portion covered per hour**	**-**	**-**
1	Machine Architecture of SIC and SIC/XE	CO1	L2
2	Basic assembler functions	CO1	L2
3	Machine dependent assembler features,	CO1	L2
4	Machine independent assembler features,	CO1	L2
5	Assembler design options.	CO1	L3
6	Basic macro processor functions	CO1	L2
7	Machine independent macro processor features	CO1	L3
8	Macro processor design options,	CO1	L2
9	Macro processor design	CO1	L3
10	Implementation examples	CO1	L2
c	**Application Areas**		
-	Students should be able employ / apply the Module learnings to . . .		
1	Design Macro Processor	CO1	L3
2	Design Assemblers	CO1	L3
d	**Review Questions**		
1	What are the Registers used in SIC machine architecture	1	L1
2	Bring out difference between system software and application software with example	CO1	L3
3	Write SIC/XE program copy array A of 100 words to array B of same size	CO1	L3
4	Generate Machine code for the following SIC/XE program COPY START 0000 CLOOP +JSUB RDREC LDA LENGTH COMP ZERO JEQ EXIT J CLOOP EXIT STA BUFFER LDA THREE STA TOTAL_LENGTH RSUB BUFFER RESW 100 EOF BYTE C 'EOF' ZERO WORD 0 THREE WORD 3 LENGTH RESW 1 TOTAL_LENGTH RESW 1 RDREC LDX ZERO MNEMONICS JSB :A0 LDA:80 LDX:60 STA:50 COMP:90 RSUB:4C JEQ:B0 J=B8	CO1	L3

5	Explain the Data Structures used in Assembler design	CO1	L2
6	Write and explain algorithm of pass 1 of two pass assembler	CO1	L3
7	Write and explain algorithm of pass 2 of two pass assembler	CO1	L3
8	explain SIC/XE machine architecture formats and all addressing modes by indicating setting of different flag bits	CO1	L2
9	Compare two pass assembler with one pass assembler . how forward references are handled in one pass assembler	CO1	L3
10	Assuming the following symbol table definitions BUFFER RELATIVE FIRST RELATIVE MAXLEN ABSOLUTE LENGTH RELATIVE BUFFEND RELATIVE Classify the following into absolute, relative or neither absolute nor relative expressions 1.BUFFER-FIRST 2. MAXLEN-1 3. BUFFER-MAXLEN 4. 2*MAXLEN-1 5. FIRST + BUFFER 6. BUFFER+4095 7. BUFFER+MAXLEN-1 8. 2*LENGTH 9. MAXLEN-BUFFER 10.FIRST-BUFFER+BUFFEND	CO1	L3
11	Explain the following a. Conditional macro expansion b.Concatenation of macro parameters c.Keyword macro parameters **d.Generation of unique labels.**	CO2	L2
e	**Experiences**	-	-
1		CO1	L2

Module – 2

Title:	**Loaders and Linkers**	Appr Time:	10 Hrs
a	***Course Outcomes***	**CO**	**Blooms**
1	Illustrate working of Loader and Linker	CO2	**L2,L3**
b	***Course Schedule***	-	-
Class No	**Portion covered per hour**	**-**	**-**
11	Basic Loader Functions, Machine Dependent Loader Features,	CO2	L2
12	Loader Design Options	CO2	L3
13	Implementation Examples.	CO2	L2
14	**System File and Library Structure**: Introduction	CO2	L2
15	Library And File Organization,	CO2	L2
16	Design Of A Record Source Program File Structure,	CO2	L2
17	Object Code, Object File,	CO2	L2
18	Object File Structure	CO2	L2

19	Executable File, Executable File Structure,	CO2	L2
20	Libraries, Image File Structure .	CO2	L2
c	**Application Areas**	**-**	**-**
-	Students should be able employ / apply the Module learnings to . . .	-	-
1	Design Loaders and Linkers	CO2	L3
2	Design File management module of operating systems	CO2	L2
d	**Review Questions**	-	-
-			
12	Explain detailed design of linking and relocating loader	CO2	L2
13	Explain various data structures used for linking loader	CO2	L2
14	Explain algorithm for simple bootstrap loader	CO2	L3
15	Explain dynamic linking with suitable programs	CO2	L3
16	Write algorithm for absolute loader	CO2	L3
17	Write algorithm for linking loader	CO2	L3
18	Explain relocation w.r.t. loader.	CO2	L2
19	Explain bitmask with an example.	CO2	L2
20	Explain program linking with an example.	CO2	L3
e	**Experiences**	-	-
1			

E1. CIA EXAM – 1

a. Model Question Paper - 1

Crs Code:	17CS63	Sem:	VI	Marks:	30	Time:	75 minutes
Course:	System Software and Compiler Design						

-	-	**Note: Answer all questions, each carry equal marks. Module : 1, 2**	**Marks**	**CO**	**Level**
1	a	Write SIC/XE program copy array A of 100 words to array B of same size	7	CO1	L3
	b	Explain SIC/XE machine architecture formats and all addressing modes by indicating setting of different flag bits	8	CO1	L2
	c				
		OR			
2	a	Explain the Data Structures used in Assembler design	4	CO2	L2
	b	Write and explain algorithm of pass 1 of two pass assembler	6	CO2	L3
	c	Explain the following a. Conditional macro expansion b.Concatenation of macro parameters c.Keyword macro parameters d.Generation of unique labels.	5	CO2	L2
	d				
3	a	Explain detailed design of linking and relocating loader	7	CO3	L3
	b	Explain various data structures used for linking loader	8	CO3	L3
	c				
		OR			
4	a	Explain algorithm for simple bootstrap loader	5	CO3	L2
	b	Explain dynamic linking with suitable programs	5	CO3	L2
	c	Write algorithm for absolute loader	5	CO3	L3

	d				

b. Assignment -1

Model Assignment Questions							
Crs Code:	17CS63	Sem:	VI	Marks:	5	Time:	90 – 120 minutes
Course:	System Software and Compiler Design			Module : 1, 2			

SNo	Assignment Description	Marks	CO	Level
1	What are the Registers used in SIC machine architecture	5	CO1	L2
2	Bring out difference between system software and application software with example	5	CO1	L3
3	Write SIC/XE program copy array A of 100 words to array B of same size		CO1	L3
4	Generate Machine code for the following SIC/XE program COPY START 0000 CLOOP +JSUB RDREC LDA LENGTH COMP ZERO JEQ EXIT J CLOOP EXIT STA BUFFER LDA THREE STA TOTAL_LENGTH RSUB BUFFER RESW 100 EOF BYTE C 'EOF' ZERO WORD 0 THREE WORD 3 LENGTH RESW 1 TOTAL_LENGTH RESW 1 RDREC LDX ZERO MNEMONICS JSB :A0 LDA:80 LDX:60 STA:50 COMP:90 RSUB:4C JEQ:B0 J=B8	5	CO1	L3
5	Explain the Data Structures used in Assembler design		CO1	L2
6	Write and explain algorithm of pass 1 of two pass assembler		CO1	L3
7	Write and explain algorithm of pass 2 of two pass assembler		CO1	L3
8	explain SIC/XE machine architecture formats and all addressing modes by indicating setting of different flag bits		CO1	L3
9	Compare two pass assembler with one pass assembler . how forward references are handled in one pass assembler		CO1	L3
10	Assuming the following symbol table definitions BUFFER RELATIVE FIRST RELATIVE MAXLEN ABSOLUTE LENGTH RELATIVE BUFFEND RELATIVE Classify the following into absolute, relative or neither absolute nor relative expressions 1.BUFFER-FIRST 2. MAXLEN-1 3. BUFFER-MAXLEN 4. 2*MAXLEN-1 5. FIRST + BUFFER 6. BUFFER+4095 7. BUFFER+MAXLEN-1 8. 2*LENGTH		CO1	L3

	9. MAXLEN-BUFFER 10.FIRST-BUFFER+BUFFEND			
11	Explain various data structures used in design of macroprocessor		CO1	L2
12	What is literal ? differentiate literals from immediate data		CO1	L2
13	What is program block and how does assembler handles program block		CO1	L2
14	List machine independent macroprocessor features. Explain any two with an example.		CO2	L2
15	What is the basic function of macroprocessor ? Explain various data structures used in design one pass macroprocessor		CO2	L3
16	What are control sections. How they are processed ?		CO2	L2
17	What is Relocation ? explain how modification record is used in relocation of a program		CO2	L3
18	Briefly explain SIC assembler directives with examples		CO1	L3
19	Write a sequence of instructions for SIC to clear a 20 byte string to all blanks.	6	CO1	L2
20	With an example, explain simple I/O operation of SIC/XE?	5	CO1	L2
21	What are the fundamental functions of assembler? With an example, give the list of assembler directives?	6	CO1	L2
22	what is program relocation? Explain the problem associated with it and solutions?	6	CO1	L2
23	Give the format of the following a. Header record b. Text record c. End record d. Modification record	8	CO1	L2
24	Explain the following a. NAMTAB b. DEFTAB c. ARGTAB	6	CO1	L2
25	Explain the following a. Conditional macro expansion b.Concatenation of macro parameters c.Keyword macro parameters d.Generation of unique labels.	8	CO1	L2
26	Explain Recursive Macro Expansion.	7	CO1	L2
27	Explain general purpose macro processors.	6	CO1	L2
28	Explain detailed design of linking and relocating loader	5	CO2	L4
29	Explain various data structures used for linking loader	5	CO2	L2
30	Explain algorithm for simple bootstrap loader	5	CO2	L3
31	Explain dynamic linking with suitable programs	7	CO2	L2
32	Write algorithm for absolute loader	8	CO2	L3
33	Write algorithm for linking loader	7	CO2	L3
34	Explain relocation w.r.t. loader.	8	CO2	L2
35	Explain bitmask with an example.	5	CO2	L2
36	Explain program linking with an example.	7	CO2	L2
37	Write the algorithm for pass 1 of an linking loader.	8	CO2	L3
38	Write the algorithm for pass 2 of an linking loader.	8	CO2	L3
39	Explain linkage editors	8	CO2	L2
40	Explain CSADDR, PROGADDR, ESTAB.	6	CO2	L2

D2. TEACHING PLAN - 2

Module – 3

Title:	Lexical Analysis	Appr Time:	10 Hrs
a	***Course Outcomes***	**CO**	**Blooms**
-	At the end of the topic the student should be able to . . .	-	**Level**
	Explain Lexical analyzers	CO3	L3
b	***Course Schedule***		
Class No	**Portion covered per hour**	**-**	**-**
21	Introduction Lexical analysis	CO3	L2
22	Alphabets And Tokens In Computer Languages	CO3	L2
23	Representation of tokens	CO3	L2
24	Token Recognition	CO3	L2
25	Finite Automata,	CO3	L2
26	Implementation of lexical Analyser	CO3	L3
27	Specifications of Tokens	CO3	L2
28	Recognition of Tokens	CO3	L2
29	Input Buffering	CO3	L2
30	Error Recovery.	CO3	L2
c	**Application Areas**	**-**	**-**
-	Students should be able employ / apply the Module learnings to . . .	-	-
	Design and develop Lexical Analysis Tools	CO3	L3
d	**Review Questions**	-	-
-	The attainment of the module learning assessed through following questions	-	-
1	Show the translations for assignment statement "position = initial+ rate * 60" in various phases of Compiler. Clearly indicate the output of each phase.	CO3	L3
2	Describe Language Processors	CO3	L2
3	Show the translations for assignment statement "position = initial+ rate * 60" in various phases of Compiler. Clearly indicate the output of each phase.	CO3	L3
4	Explain three types of software productivity tools.	CO3	L2
5	Give transition diagram for unsigned numbers.	CO3	L3
6	Show that the following grammar is ambiguous. Write equivalent unambiguous grammar for the same. E->E+E / E*E / (E) / id	CO3	L4
7	Describe Language Processors	CO3	L2
8	Show the translations for assignment statement "position = initial+ rate * 60" in various phases of Compiler. Clearly indicate the output of each phase.	CO3	L2
9	Give transition diagram for unsigned numbers.	CO3	L3
10	Show that the following grammar is ambiguous. Write equivalent unambiguous grammar for the same. E->E+E / E*E / (E) / id	CO3	L3
e	**Experiences**	-	-
1			

Module – 4

Title:	Syntax Analyser	Appr Time:	10 Hrs
a	***Course Outcomes***	**CO**	**Blooms**
-	At the end of the topic the student should be able to . . .	-	**Level**
1	Design Parsers	CO4	L3
b	***Course Schedule***		
Class No	**Portion covered per hour**	**-**	**-**

31	**Syntax Analysis:** Introduction,	CO4	L2
32	Role Of Parsers,	CO4	L2
33	Context Free Grammars	CO4	L2,L3
34	Context Free Grammars	CO4	L3
35	Top Down Parsers,	CO4	L2,L3
36	Top Down Parsers,	CO4	L2,L3
37	Bottom-Up Parsers	CO4	L2,L3
38	Bottom-Up Parsers	CO4	L2,L3
39	Operator-Precedence Parsing	CO4	L2,L3
40	Operator-Precedence Parsing	CO4	L2,L3
c	**Application Areas**	**-**	**-**
-	Students should be able employ / apply the Module learnings to . . .	-	-
1	Design Parsers	CO4	L3
d	**Review Questions**	-	-
-	The attainment of the module learning assessed through following questions	-	-
1	Construct predictive parsing table by making necessary changes to the grammar given below and sow the parsing of the string Id+id*id(LL parsing) E ->E+T/T T->T*F/F F-> (E) / id	CO4	L3
2	What is shift reduce parser ? explain its actions and conflicts by taking an example.	CO4	L3
3	Design SLR parser for the following grammar by computing LR(0) items and show the parsing of the string ((a))	CO4	L3
4	Construct CLR parser by finding LR(1) items for the following grammar S ->AA A->aA / b	CO4	L3
5	Define inherited and Synthesized attributes. Give Examples	CO4	L3
6	Define syntax directed definition for simple type declaration	CO4	L3
e	**Experiences**	-	-
1			
2			

E2. CIA EXAM – 2

a. Model Question Paper - 2

Crs Code:	17CS63	Sem:	VI	Marks:	30	Time:	75 minutes
Course:	System Software and Compiler Design						

-	-	**Note: Answer all questions, each carry equal marks. Module : 3, 4**	**Marks**	**CO**	**Level**
1	a	Show the translations for assignment statement "position = initial+ rate * 60" in various phases of Compiler. Clearly indicate the output of each phase.	5	CO3	L3
	b	Explain three types of software productivity tools.	4	CO3	L2
	c	how that the following grammar is ambiguous. Write equivalent unambiguous grammar for the same. E->E+E / E*E / (E) / id	6	CO3	L4
	d				
2	a	Consider the production S->aAb A->cd / C	7	CO4	

		show that the recursive descent parser fails for the input string "acdb", also explain recursive descent algorithm			
	b	Consider the following grammar S ->L = R/R L->*R/id R-> L i)obtain LR(0) items ii) compute FIRST and FOLLOW sets iii) obtain SLR parsing table iv) check whether the given grammar is SLR or not	8	CO4	L4
	c				
	d				
3		Construct predictive parsing table by making necessary changes to the grammar given below and sow the parsing of the string Id+id*id(LL parsing) E ->E+T/T T->T*F/F F-> (E) / id	7	CO4	L1
		Find FIRST and FOLLOW set for the grammar given below i) stmt_sequence->stmt stmt_sequence' stmt_sequence'->; stmt stmt_sequence' / ε stmt ->s ii) S->, GH; G->aF F->bF / ε H -> KL K ->m / ε L -> n / ε	8	CO4	L2
		OR			
4		Design SLR parser for the following grammar by computing LR(0) items and show the parsing of the string ((a))	8	CO4	L3
		Construct CLR parser by finding LR(1) items for the following grammar S ->AA A->aA / b	7	CO4	L3

b. Assignment – 2

Model Assignment Questions							
Crs Code:	17CS63	Sem:	6	Marks:	5	Time:	
Course:	Systems software and compiler design						

SNo	Assignment Description	Marks	CO	Level
1	Show the translations for assignment statement "position = initial+ rate * 60" in various phases of Compiler. Clearly indicate the output of each phase.	5	CO3	L3
2	Explain three types of software productivity tools.	3	CO3	L3
3	Give transition diagram for unsigned numbers.	4	CO3	L3
4	Show that the following grammar is ambiguous. Write equivalent unambiguous grammar for the same. E->E+E / E*E / (E) / id	6	CO3	L3
5	Describe Language Processors	4	CO3	L3
6	Consider the following grammar S ->L = R/R L->*R/id R-> L i)obtain LR(0) items ii) compute FIRST and FOLLOW sets iii) obtain SLR parsing table	8	CO4	L3

	iv) check whether the given grammar is SLR or not			
7	Define left recursive grammar Eliminate left recursion from the following grammar E –>E+T/T, T–>T*F/F, F–> (E) / id	6	CO4	L3
8	What are the problems with top down parser? Write Recursive Descent Parser for the grammar S–>cAd, A–>ab/a	8	CO4	L3
9	Describe algorithm for elimination of left recursion.	4	CO3	L3
10	List algebraic laws of regular expression	4	CO3	L3
11	Explain various phases of compiler with neat diagram	5	CO3	L3
12	Write transition diagram along with the program code to recognize token below 1} RELOP 2) Unsigned number	6	CO3	L3
13	Write rules for constructing FIRST and FOLLOW sets	4	CO4	L3
14	Construct predictive parsing table by making necessary changes to the grammar given below and show the parsing of the string Id+id*id(LL parsing) E –>E+T/T T–>T*F/F F–> (E) / id	8	CO4	L3
15	What is shift reduce parser ? explain its actions and conflicts by taking an example.	6	CO4	L2
16	Design SLR parser for the following grammar by computing LR(0) items and show the parsing of the string ((a))	6	CO4	L3
17	Construct CLR parser by finding LR(1) items for the following grammar S –>AA A–>aA / b	6	CO4	L3
18	Define inherited and Synthesized attributes. Give Examples	5	CO4	L2
19	Define syntax directed definition for simple type declaration	6	CO4	L2
20	Given the grammar S–>(L) / α L–>L,S / S i)Make necessary changes to make it suitable for LL(1) parsing ii)Compute FIRST and FOLLOW sets iii)Construct the predictive parsing table. iv) Show the moves made by the parser on the input (a,(a,a))	8	CO4	L3
21	Write recursive descent parser for the grammar S–>cAd A–>ab / b and for the input “cad” trace the parser.	8	CO4	L3
22	Show that the following grammar is not LL(1) without constructing parsing table. S–>iCtSS' / α S'–>eS / ε C–>b	5	CO4	L3
23	What is meant by handle pruning ? show the working of shift reduce parse for accepting id+id*id, considering the grammar E –>E+T/T T–>T*F/F F–> (E) / id	4	CO4	L4
24	For the following grammar S –>0S1/ 01, indicate the handle in the following right sentential form 00001111	4	CO4	L3
25	Write algorithm to eliminate left recursion from the grammar, also give the syntax of the production	5	CO4	L3
26	Consider the production S–>aAb A–>cd / C show that the recursive descent parser fails for the input string “acdb”, also	7	CO4	L3

	explain recursive descent algorithm			
27	Find FIRST and FOLLOW set for the grammar given below i) stmt_sequence->stmt stmt_sequence' stmt_sequence'->; stmt stmt_sequence' / ε stmt ->s ii) S->, GH; G->aF F->bF / ε H -> KL K ->m / ε L -> n / ε	8	CO4	L3

D3. TEACHING PLAN - 3

Module – 5

Title:	Syntax Directed Translation, Intermediate code generation	Appr Time:	10 Hrs
a	***Course Outcomes***	**CO**	**Blooms**
-	At the end of the topic the student should be able to . . .	-	**Level**
	Discuss Syntax Directed Translation and Code Generators	CO5	l3
b	***Course Schedule***	-	-
Class No	**Portion covered per hour**	**-**	**-**
41	Syntax Directed Translation	CO5	L3
42	Syntax-directed translation schemes	CO5	L3
43	Variants of syntax trees;	CO5	L2
44	Three-address code	CO5	L3
45	Translation of expressions	CO5	L4
46	Intermediate code generation,	CO5	L3
47	Control flow; Back patching	CO5	L3
48	Switch statements; Procedure calls	CO5	L3
49	Code Generation	CO5	L3
50	A Simple Code Generator	CO5	L3
c	**Application Areas**	**-**	**-**
-	Students should be able employ / apply the Module learnings to . . .	-	-
1	To design semantic analysis and code generation of compilers	CO5	L3
d	**Review Questions**	-	-
-	The attainment of the module learning assessed through following questions	-	-
1	Construct DAG and three address code for the following expression a+a*(b-c)+(b-c)*d	CO5	L2
2	Discuss various issues in design of code generator	CO5	L2
3	Explain the following with example i)Quadruples ii) triples iii)indirect triples	CO5	L2

4	Generate three address code to the following statement
	Switch(ch)
	{
	Case 1 :a+b; break;
	Case 2 :a-b; break;
	}
		CO5	L2

5	For the following program segment For i=1 to 10 For j= 1 to 10 a[I,j]=0.0 For i=1 to 10 a[I,j]=1.0	CO5	L2

	generate immediate code and identify basic blocks		

E3. CIA EXAM – 3

a. Model Question Paper - 3

Crs Code:	17CS63	Sem:	VI	Marks:	30	Time:	75 minutes
Course:	System Software and Compiler Design						

-	-	**Note: Answer all questions, each carry equal marks. Module : 5**	**Marks**	**CO**	**Level**

1	a	Generate three address code to the following statement
		Switch(ch)
		{
		Case 1 :a+b; break;
		Case 2 :a-b; break;
		}
			5	CO5	L2

	b	With neat diagram describe the general structure of an activation record	4	CO5	L2
	c	Define syntax directed definition for simple type declaration	6	CO5	L2
	d			CO5	
2	a	Construct DAG and three address code for the following expression a+a*(b-c)+(b-c)*d	7	CO5	L3
	b	Discuss various issues in design of code generator	4	CO5	L2
	c	Explain the strategy for reducing fragmentation in heap memory	4	CO5	L2
		OR			
	a	Construct predictive parsing table by making necessary changes to the grammar given below and sow the parsing of the string Id+id*id(LL parsing) E ->E+T/T T->T*F/F F-> (E) / id	7	CO5	L3
3	a	Construct DAG and three address code for the following expression a+a*(b-c)+(b-c)*d	5	CO5	L3
	b	Discuss various issues in design of code generator	3	CO5	L3
	c	Explain the following with example i)Quadruples ii) triples iii)indirect triples	4	CO5	L3
		OR			
4	a	Write syntax directed definition for flow control statements	5	CO5	L3
	b	With neat diagram describe the general structure of an activation record	4	CO5	L3

b. Assignment – 3

Model Assignment Questions							
Crs Code:	17CS63	Sem:	6	Marks:	10	Time:	
Course:	Systems software and Compiler Design						

SNo	**Assignment Description**	**Marks**	**CO**	**Level**
1	Construct DAG and three address code for the following expression a+a*(b-c)+(b-c)*d	5	CO5	L3
2	Discuss various issues in design of code generator	3	CO5	L3
3	Explain the following with example i)Quadruples ii) triples iii)indirect triples	4	CO5	L3
4	Generate three address code to the following statement Switch(ch) { Case 1 :a+b; break; Case 2 :a-b; break;	5	CO5	L4

	}			
5	For the following program segment For i=1 to 10 For j= 1 to 10 a[I,j]=0.0 For i=1 to 10 a[I,j]=1.0 generate immediate code and identify basic blocks	5	CO5	L4
6	Construct DAG and generate 3-address code for the following expressions. **x + x*(y-z) + (y + z)*w**	5	CO5	L3
7	Write syntax directed definition for flow control statements	5	CO5	L3
8	With neat diagram describe the general structure of an activation record	4	CO5	L3
9	Explain briefly the performance metrics to be considered while designing garbage collector.	4	CO5	L3
10	Explain the strategy for reducing fragmentation in heap memory	3	CO5	L3
11	Explain various issues in the design of code generator		CO5	L3
12	By considering array of type int [3][3] write syntax directed translation with semantic rules	8	CO5	L3
13	Give SDD for simple desk calculator and draw annotated parse tree for expression (3+4)*(5+6)	5	CO5	L3
14	Explain the concept of syntax directed definition	4	CO5	L3
15	Consider the contex free grammar given below S->EN E ->E+T/E-T/T T->T*F / T/F / F F-> (E) / digit N ->; i) obtain SDD for the above grammar ii) Annotated parse tree for the input string 5*6+7	8	CO5	L3
16	Define i)synthesized attribute ii)Inherited attribute	5	CO5	L3
17	Generate 3 – address statements for the following programming construct and obtain the basic blocks for generated code i=1 do sum =sum + a[i] * b[i] i = i+1 while (i<=20)	6	CO5	L3

F. EXAM PREPARATION

1. University Model Question Paper

Course:	System Software and Compiler Design					Month / Year	May /2018
Crs Code:	17CS63	Sem:	VI	Marks:	80	Time:	180 minutes

Module	Note	Answer all FIVE full questions. All questions carry equal marks.	Marks	CO	Level
1	a	Define System software. Distinguish between system software and application software	8	CO1	L2
	b	List out registers used in SIC/XE Machine architecture along with their use	8	CO1	L2
		OR			
1	a	Explain the data structure and pass-1 algorithm of SIC assembler	8	CO1	L2
	b	Define Macro. Give the features of macroprocessor nd explain the data structures used in macro processor	8	CO1	L2

2	a	What is loader? What are the advantages and disadvantages? Explain boot strap algorithm	8	CO2	L2
	b	Enlist any four different Loader option commands	4	CO2	L2
	c	Define 1) linking loader 2) dynamic linking	4	CO2	L1
		OR			
2	a	Explain the working of linkage editor and linking loader	8	CO2	L2
	b	With the data structure explain pass 2 algorithm for linking loader	8	CO2	L3
3	a	What is compiler ? Explain various phases of compiler with the help of neat diagram.	10	CO3	L2
	b	Explain the concept of input buffering In the lexical analysis	6	CO3	L2
		OR			
3	a	Construct the transition diagram to recognize the tokens given below i)identifiers ii) Relational operators iii) unsigned number	8	CO3	L3
	b	Create lexical analyser with LEX and explain the structure of LEX programs	8	CO3	L3
4	a	What are top down parser? What are key problems in top down parsing	8		L2
	b	Explain the Ambiguity in arithmetic expression. What is the ambiguity in parsing 2+3*4 ? explain the solution for it.	8		L3
		OR			
4	a	What is meant by handle pruning ? How it helps on shift reduce parsing ?	8	CO4	L2
	b	Form action goto table for the following grammar S ->Aa/bAc/Ba/bBa A->d B-> d	8	CO4	L3
5	a	Discuss L-attribute and S-attribute wrt SDD	4	CO5	L2
	b	Construct syntax tree for x*y-5+z	4	CO5	L3
	c	Explain the following with example i)Quadruples ii) triples iii)indirect triples	8	CO5	L2
		OR			
5	a	briefly explain the main issues in code generation	8	CO5	L2
	b	Generate code for the expression x=(a+b) * (c-d)	8	CO5	L3

2. SEE Important Questions

Course:	System Software and Compiler Design					Month / Year	May /2018
Crs Code:	17CS63	Sem:	6	Marks:	60	Time:	180 minutes

	Note	Answer all FIVE full questions. All questions carry equal marks.	-	-	
Module	Qno.	Important Question	**Marks**	**CO**	**Year**
1	1	Bring out difference between system software and application software with example	5	CO1	2016
	2	Write SIC/XE program copy array A of 100 words to array B of same size	5	CO1	2015
	3	explain SIC/XE machine architecture formats and all addressing modes by indicating setting of different flag bits	10	CO1	2018
2	1	Generate Machine code for the following SIC/XE program COPY START 0000 CLOOP +JSUB RDREC LDA LENGTH COMP ZERO JEQ EXIT J CLOOP EXIT STA BUFFER LDA THREE STA TOTAL_LENGTH RSUB BUFFER RESW 100	10	CO1	2017

		EOF BYTE C 'EOF' ZERO WORD 0 THREE WORD 3 LENGTH RESW 1 TOTAL_LENGTH RESW 1 RDREC LDX ZERO MNEMONICS JSB :A0 LDA:80 LDX:60 STA:50 COMP:90 RSUB:4C JEQ:B0 J=B8			
	2	Write and explain algorithm of pass 2 of two pass assembler	10	CO1	2017
3	1	Write algorithm for absolute loader	6	CO2	2018
	2	What are control sections ? How they are processed ?	6	CO1	2017
	3	Explain two design options of loader	7	CO2	2016
	4	What is loader :? what are advantages and disadvantages ? Explain bootstrap loader with example	10	CO2	2016
4	1	Construct predictive parsing table by making necessary changes to the grammar given below and show the parsing of the string Id+id*id(LL parsing) E ->E+T/T T->T*F/F F-> (E) / id	10	CO4	2018
	2	Consider the contex free grammar given below S->EN E ->E+T/E-T/T T->T*F / T/F / F F-> (E) / digit N ->; i) obtain SDD for the above grammar ii) Annotated parse tree for the input string 5*6+7	8	CO5	2017
	3	Find FIRST and FOLLOW set for the grammar given below i) stmt_sequence->stmt stmt_sequence' stmt_sequence'->; stmt stmt_sequence' / ε stmt ->s ii) S->, GH; G->aF F->bF / ε H -> KL K ->m / ε L -> n / ε	8	CO4	2016
5	1	Construct DAG and three address code for the following expression a+a*(b-c)+(b-c)*d	6	CO6	2016
	2	Explain the following with example i)Quadruples ii) triples iii)indirect triples	6	CO5	2015
	3	Construct DAG and generate 3-address code for the following expressions. **x + x*(y-z) + (y + z)*w**	7	CO5	2016
	4	Write syntax directed definition for flow control statements	6	CO5	2014

Course Outcome Computation

Academic Year:

Odd / Even semester

INTERNAL TEST					**T1**						**T2**	
Course Outcome	**CO1**		**CO2**		**CO3**		**CO4**		**CO5**		**CO6**	
QUESTION NO	**Q1**	**LV**	**Q2**	**LV**	**Q3**	**LV**	**Q1**	**LV**	**Q2**	**LV**	**Q3**	**LV**
MAX MARKS	10	-	10	-	10	-	10	-	10	-	10	-
USN-1	5	2	10				10	3	9	3	4	1
USN-2	5	2	8	3								
USN-3	7	3	7	3	10	3	8	3	8	3	5	2
USN-4					4	1	10	3	8	3	6	2
USN-5	8	3	6	2	9	3	10	3	8	3		
USN-6							10	3	9	3	4	1
Average CO Attainment		2.5		2.75		2.33		3		3		1.5

LV Threshold : 3:>60%, 2:>=50% and <=60%, 1: <=49%
CO1 Computation :(2+2+2+3)/4 = 10/4=2.5

PO Computation

Program Outcome	PO1		PO3		PO3		PO1		PO12		PO12	
Weight of CO - PO	3		1		3		2		2		3	
Course Outcome	CO1		CO2		CO3		CO4		CO5		CO6	
Test/Quiz/Lab	T1						T2					
QUESTION NO	Q1	LV	Q2	LV	Q3	LV	Q1	LV	Q2	LV	Q3	LV
MAX MARKS	10	-	10	-	10	-	10	-	10	-	10	-
USN-1	5	2	10	3			10	3	9	3	4	1
USN-2	5	2	8	3								
USN-3	7	3	7	3	10	3	8	3	8	3	5	2
USN-4					4	1	10	3	8	3	6	2
USN-5	8	3	6	2	9	3	10	3	8	3		
USN-6							10	3	9	3	4	1
Average CO Attainment		2.5		2.75		2.33		3		3		1.5