Lesson Plan

Branch: Computer Engineering
Semester: VI   **Year: 2022-23**

Course Title: System Programming Compiler Construction (CSC601)	SEE: 3 Hours – Theory
Total Contact Hours: 39 Hours	Duration of SEE: 3 Hrs
SEE Marks: 80 (Theory) + 20 (IA)	
Lesson Plan Author: Prof. Supriya Kamoji	Date:
Checked By:	Date:

Prerequisites: TCS

Syllabus:

Module	Content	Hrs
1	**Introduction to System Software** Concept of System Software, Goals of system software's, system program and system programming, Introduction to various system programs such as Assembler, Macro processor, Loader, Linker, Compiler, Interpreter, Device Drivers, Operating system, Editors, Debuggers	2
2	**Assemblers** Elements of Assembly Language programming, Assembly scheme, pass structure of assembler, Assembler Design: Two pass assembler Design and single pass Assembler Design for Hypothetical / X86 family processor, data structures used.	7
3	**Macros & Macro processors** Introduction, Macro definition and call, Features of Macro facility: Simple, parameterized, conditional and nested. Design of single pass macro processor, data structures used	6
4	**Loaders and Linkers** Introduction, functions of loaders, Relocation and Linking concept, Different loading schemes: Relocating loader, Direct Linking Loader, Dynamic linking and loading.	6
5	**Compilers: Analysis Phase** Introduction to compilers, Phases of compilers: **Lexical Analysis**- Role of Finite State Automata in Lexical Analysis, Design of Lexical analyser, data structures used . **Syntax Analysis**- Role of Context Free Grammar in Syntax analysis, Types of Parsers: Top down parser- LL(1), Bottom	10

	up parser- Operator precedence parser, SLR **Semantic Analysis**, Syntax directed definitions.	
6	**Compilers: Synthesis Phase** **Intermediate Code Generation**: Types of Intermediate codes: Syntax tree, Postfix notation, Three address codes: Triples and Quadruples. **Code Optimization**: Need and sources of optimization, Code optimization techniques: Machine Dependent and Machine Independent. **Code Generation**: Issues in the design of code generator, code generation algorithm. Basic block and flow graph.	**8**

Course Outcomes (CO):

On successful completion of course learner will be able to:

Identify the relevance of different system programs.	
CSC601.1	Explain the design of two pass assembler .
CSC601.2	Explain the design of two pass macroprocessor.
CSC601.3	Distinguish between different loaders and linkers schemes.
CSC601.4	Explore analysis phase of the compiler.
CSC601.5	Describe synthesis phase of compiler

CO-PO Mapping: (BL – Blooms Taxonomy, C – Competency, PI – Performance Indicator)

CO	BL	C	PI	PO	Mapping
CSC601.1.	1, 2, 3	1.3 2.1 3.2	1.3.1 2.1.3 3.2.1	PO1 PO2 PO3	1 1 1
CSC601.2.	1, 2, 3	1.3 2.1 3.2	1.3.1 2.1.3 3.2.1	PO1 PO2 PO3	1 1 1
CSC601.3.	1,2	1.3	1.3.1	PO1	1
CSC601.4.	1,2,3	1.3 2.1 3.3	1.3.1 2.1.3 3.3.1	PO1 PO2 PO3	1 1 1
CSC601.5.	1,2,3	1.3 2.1	1.3.1 2.1.3	PO1 PO2	1 1

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CSC601.1	1	1	1									
CSC601.2	1	1	1									
CSC601.3	1											
CSC601.4	1	1	1									
CSC601.5	1	1	1									

CO-PSO Mapping:

CO	BL	C	PI	PO	Mapping
CSC601.4.	2	1.1	1.1.3	PSO1	1
CSC601.5.	2	1.1	1.1.3	PSO1	1

	PSO1	PSO2
CSC602.1.	--	--
CSC602.2.	--	--
CSC602.3.	--	--
CSC602.4.	--	1
CSC602.5.	--	1

CO Measurement Weightages for Tools:

Course Outcomes	*Direct Tools (80 %)*									*Indirect Method (20%)*
	Unit Tests		Assignments			Quizzes			End Sem Exam	Course exit survey
	1	2	1	2	3	1	2	3		
CSC601.1		20%	20%				--		60%	100%
CSC601.2		20%		20%			-		60%	100%
CSC601.3		15%		-	-		--	25%	60%	100%
CSC601.4	20%	-	-	-	20%	-	-		60%	100%
CSC601.5	20%			-		20%			60%	100%

Attainment:

CO CSC601.1:
Direct Method
$$A_{\text{CSC601.1}D} = 0.2 * Test1 + 0.2 * Assignment + 0.6 * SEE\_Theory$$
Final Attainment:
$$A_{\text{CSC602.1}} = 0.8 * A_{\text{CSC602.1}D} + 0.2 * A_{\text{CSC602.1}I}$$

CO CSC601.2:
Direct Method
$$A_{\text{CSC602.2}D} = 0.2 * Test1 + 0.2 * Assignment + 0.6 * SEE\_Theory$$
Final Attainment:
$$A_{\text{CSC602.2}} = 0.8 * A_{\text{CSC602.2}D} + 0.2 * A_{\text{CSC602.2}I}$$

CO CSC601.3:
Direct Method
$$A_{\text{CSC602.3}D} = 0.15 * Test1 + 0.25 * Quizzes + 0.6 * SEE\_Theory$$
Final Attainment:
$$A_{\text{CSC602.3}} = 0.8 * A_{\text{CSC602.3}D} + 0.2 * A_{\text{CSC602.3}I}$$

CO CSC601.4:
Direct Method

$$A_{\text{CSC602.4}D} = 0.2 * Test2 + 0.2 * Assignment + 0.6 * SEE\_Theory$$

Final Attainment:

$$A_{\text{CSC602.4}} = 0.8 * A_{\text{CSC602.4}D} + 0.2 * A_{\text{CSC602.4}I}$$

CO CSC601.5:

Direct Method

$$A_{\text{CSC602.5}D} = 0.2 * Test2 + 0.2 * Quizzes + 0.6 * SEE\_Theory$$

Final Attainment:

$$A_{\text{CSC602.5}} = 0.8 * A_{\text{CSC602.5}D} + 0.2 * A_{\text{CSC602.5}I}$$

Course Level Gap (if any): Nil

Content beyond Syllabus: Nil

Lecture Plan:

Module	Contents	Hours	Planned date	Actual date	Content Delivery Method	Remark
1	Introduction to System Software	1	**9-01-23**		PPT	Case study on different compilers
5	Introduction to compilers, Phases of compilers, Lexical Analysis- Role of Finite State Automata in Lexical Analysis	10	10-01-23		PPT	
	Types of Parsers: Top-down parser		11-01-23		PPT	
	First set		16-01-23		PPT & Board	
	Follow set		17-01-23		PPT & Board	
	Design of Predictive parser		18-01-23		Board	
	Design of Predictive parser and LL(1) Grammar		23-01-23		PPT & Board	
	Design of recursive descent parser		27-01-23		Board	
	Bottom-up parser- Operator precedence parser		25-01-23		Board	
	Design of SLR Parser,		30-01-23			
	Examples of LR(0)		02-02-23			Assignment 1 on module 5
6	Syntax directed definitions.	8	03-02		PPT & Board	Quiz 1 on module 6
	Intermediate Code Generation: Types of Intermediate codes: Syntax tree, Postfix notation, Three address codes: Triples and Quadruples.		06		Board	
	Code Optimization: Need and sources of optimization		9-02-23		PPT	
	Code optimization techniques: Machine Dependent and Independent		10-02-23		PPT	
	Code Generation: Issues in the design of code generator,		13-02-23		PPT	
	Code generation algorithm.		16-02-23		PPT & Board	
	Basic block and flow graph.		17-02-23		PPT & Board	
	Activity on Code Optimization		20-02-23		PPT	Activity in Classroom

2	Elements of Assembly Language programming, Pass structure of assembler	6	23-02-23		PPT	
	Assembler Design: Two pass assembler Design		24-02			
	Assembler Design: Two pass assembler Design		09-03-23		PPT	
	Assembler Design: Two pass assembler Design		13-03-23		PPT	
	Assembler Design: Two pass assembler Design		16-03-23		PPT	
	Single pass Assembler		17-03-23		PPT	
	Difference in design of single and two pass assembler		20-03-23		PPT	Assignment 1 on module 2
3	Features of Macro facility: Simple, parameterized, conditional, and nested	4	23-03-23		PPT	
	Design of two pass macro processor, data structures used		24-03-23		PPT	Assignment 3 on module 3
	Design of two pass macro processor, data structures used		03-04-23		PPT	
	Design of two pass macro processor, data structures used		06-04-23		PPT	
6	Introduction, functions of loaders, Relocation and Linking concept	4	10-04		PPT	Quiz on module 4
	Direct Linking Loader,		13-04		PPT	
	Direct Linking Loader,		Extra		PPT	
	Dynamic linking and loading.		Extra			

Reference Books:
1. Lex and Yacc , 2nd Edition
2. Compiler construction D,M.Dhamdhere second edition MACMILLAM.
3. Compiler construction : principles and practices , Kenneth C.Louden ,CENGAGE Learning
4. System software : An introduction to system programming , Leland L. Beck, Pearson

Assessment:
Internal Assessment:
Assessment consists of two class tests of 20 marks each. The first class test is to be conducted when
approx. 40% syllabus is completed and second class test when additional 40% syllabus is completed.
Duration of each test shall be one hour.

End Semester Theory Examination:

1. Question paper will comprise of 6 questions, each carrying 20 marks.
2. The students need to solve total 4 questions.
3. Question No.1 will be compulsory and based on entire syllabus.
Remaining question (Q.2 to Q.6) will be selected from all the modules
Online resources:

- **https://nptel.ac.in/courses/106/108/106108113/**
- **https://nptel.ac.in/courses/106/105/106105190/**
- **https://www.tutorialspoint.com/compiler_design/index.htm**
- **https://www.geeksforgeeks.org/last-minute-notes-compiler-design-gq/**
- **https://pyq.ravindrababuravula.com/subject/?cs=Compiler-Design**
- **https://www.youtube.com/watch?v=oAnkQJT85Ts**
- https://www.youtube.com/watch?v=7Nb-NTGbe-Q&list=PLbRMhDVUMngcseCW7wXDvtTDemCuH80fP&index=2
- https://www.youtube.com/watch?v=xBQEqI91cJM&list=PLbRMhDVUMngcseCW7wXDvtTDemCuH80fP&index=8
- https://www.youtube.com/watch?v=IdCT0KgUZXU&list=PLbRMhDVUMngcseCW7wXDvtTDemCuH80fP&index=17
- https://www.youtube.com/watch?v=QFVorwdIZRg&list=PLbRMhDVUMngcseCW7wXDvtTDemCuH80fP&index=30

Evaluation Scheme

CIE Scheme
Internal Assessment: 20 (Average of two tests)

Internal Assessment Scheme

	Module	Lecture Hours	No. of questions in		No. of questions in SEE
			Test 1	Test 2	
5	Compilers: Analysis Phase	10	02 (10 marks)	--	3
6	Compilers: Synthesis Phase	08	02 (10 Marks)	--	3/4
2	Assembler	6		01 (8 Marks)	2
3	Macroprocessor	4	--	01 (7 Marks)	2
4	Loader and Linker	4	--	01 (5 Marks)	1

Note: Four to six questions will be set in the Test paper

 Verified by:

 Programme Coordinator   Subject Expert