LESSON PLAN

NAME OF SUBJECT INCHARGE: MR. VIKAS KR, SINGH /MR. SUMAN KALYAN KAR/ MRS. CHITRAPRIYA N.

NAME OF SUBJECT COORDINATOR: MR. VIKAS KR, SINGH /MR. SUMAN KALYAN KAR/ MRS. CHITRAPRIYA N.

NAME OF SUBJECT: COMPILER DESIGN

DEPARTMENT: CSE

SEMESTER (SECTION): VII

TOTAL NO. OF UNITS: 02

TOTAL NO. OF TOPICS: 8

HOURS ALLOTED PER WEEK: 4

MINIMUM HOURS ALLOTED FOR SEMESTER: 40 Hours

CREDITS: 04

SCHEDULE OF CLASSES:

Sl . No.	Day	Period/Time
1		
2		
3		
4		

TOTAL WORKING DAYS:

TOTAL WORKING HOURS:

Subject In-charge **Subject Coordinator** **HOD/CSE**

PROGRAM OUTCOMES

PO 1	**Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
PO 2	**Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
PO 3	**Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
PO 4	**Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
PO 5	**Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
PO 6	**The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
PO 7	**Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
PO 8	**Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
PO 9	**Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
PO 10	**Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
PO 11	**Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
PO 12	**Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Subject-Incharge **Subject Coordinator** **HOD/CSE**

PROGRAM SPECIFIC OUTCOMES (PSO)

PSO1	Students to have knowledge and expertise in at least one procedural and one object oriented programming language and should be able to analyze and compare algorithms.
PSO2	Students must have the ability to visualize and solve problems using appropriate structures and constraints adhering to existing Software Engineering standards.
PSO3	Students must be able to design and implement database solutions using current technologies.
PSO4	Students should be able to understand, troubleshoot and design computer networks, including distributed networks and wireless networks.
PSO5	Students should be aware of the design principles of Operating Systems specializing on at least one popular Operating System and System Programs. Students should have working knowledge on Advanced Computing techniques for Machine Learning and Computer Intelligence.
PSO6	Students will be able to keep pace with the technological advancement through exposure to recent and emerging trends of Computer Science ranging from Big Data, Cloud Computing, Data Analytics, Social networking, Mobile Robotics, Artificial Intelligence, Internet of Things(IoT), Augmented Reality etc.

COURSE OUTCOMES

On successful completion of this course, students will be able to:

CO1	Demonstrate competence in designing compilers and their functioning in various problem domains.
CO2	Demonstrate an ability to evaluate complex engineering problem and able to solve it using the concept of compiler design.
CO3	Demonstrate an ability to select optimal design scheme using compiler design principles.
CO4	Create tools using knowledge of compilers for compiling different programs.

Subject In-Charge	**Subject Coordinator**	**HOD/CSE**

Performance Indicator (PI)

PI 1.6.1	Apply fundamental engineering concepts to solve engineering problems.
PI 2.5.2	Identify processes/ modules/ algorithms of a computer-based system and parameters to solve a problem.
PI 2.5.3	Identify mathematical algorithmic knowledge that applies to a given problem.
PI 3.7.1	Able to perform systematic evaluation of the degree to which several design concepts meet the criteria.
PI 5.4.1	Identify modern engineering tools, techniques and resources for engineering Activities.
PI 5.4.2	Create/adapt/modify/extend tools and techniques to solve engineering problems

INTERNAL EXAMINATION DETAILS

Quiz I

Quality Indicators (QI)	Performance Indicator (PI)			Course Outcome (CO)		Program Specific Outcome (PSO)		Bloom's Taxonomy Level (BL)				Program Outcome (PO)	
Marks													
Attainment (%)													

Sessional I

QI	PI					CO			PSO			BL				PO
Marks																
Attainment (%)																

Subject In-charge **Subject Coordinator** **HOD/CSE**

Quiz II

QI	PI			CO		PSO		BL				PO	
Marks													
Attainment (%)													

Sessional II

QI	PI					CO			PSO			BL				PO	
Marks																	
Attainment (%)																	

BL – Bloom's Taxonomy Levels (1- Remembering, 2- Understanding, 3 – Applying, 4 – Analyzing, 5 – Evaluating, 6 - Creating);

Subject In-Charge **Subject Coordinator** **HOD/CSE**

SIKKIM MANIPAL INSTITUTE OF TECHNOLOGY
ODD SEMESTER, 2020
LESSON PLAN

Subject Code : CS 1703
Subject Name : COMPILER DESIGN
Teacher in-charge : V.K. SINGH / S.K.KAR / CHITRAPRIYA N.

1. **Objective:** The course is aimed at offering complete knowledge on compiler design and ends with the development of a working compiler in parts. Topics include compiler structure, symbol tables, regular expressions and languages, finite automata, lexical analysis, context-free languages, LL(1), recursive descent, LALR(1), and LR(1) parsing semantic analysis, and code generation. This will enable the learners to use formal attributed grammars for specifying the syntax and semantics of programming languages and their impact on compiler design.

2. **Scope:**This course is suitable for those students who have exposure to the basic courses like Formal languages and automata Theory, Discrete structures for computer science and programming skills.

3. **Text Books:**

   (T1) A.V. Aho, R. Sethi, J.D. Ullman, "Compilers: Principles, Techniques and Tools",Addison – Wesley. ISBN 0-321-48681-1
   (T2) Steven S. Muchnick, "Advanced Compiler Design and Implementation", Elsevier. ISBN-13: 978-1558603202

4. **Reference Books:**

   (R1) W. Appel, "Modern Compiler Implementation in C: Basic design", Cambridge Press , ISBN- 978-0521607650
   (R2) Dhamdhere, "Compiler Construction", McMillan.ISBN-10-0333904060
   (R3) A. V. Aho and J. D. Ullman, "Theory of Parsing, Translation and Compiling",Prentice Hall, ISBN-13: 978-0139145643

5. **Examination rules:**
   (i) Questions to be set having equal weightage/marks covering the entire syllabus: EIGHT (4 questions each from UNIT I and UNIT II)

(ii) Questions to be answered: FIVE (5) selecting atleast TWO from each unit

6. LECTURE MODULES:

SL. No.	Module	Learning Objective	Program Specific Outcomes(PSO)/ Course Outcomes(CO)/ Program Outcomes(PO)/PI
1	**COMPILER FUNDAMENTALS, COMPILER STRUCTURE**	Overview of Analysis-synthesis model of compilation. Discuss the various phases of compilation? What are the tools based approach to compiler construction?	PSO -2,5 CO-1,2,3 PO-1,2 PI 1.6.1,2.5.2, 2.5.2, 2.5.3,3.7.1
2	**LEXICAL ANALYSIS**	Define parser, symbol table, tokens, lexemes and patterns, regular expression definition. How transition diagrams are used for representing regular definition. Study of Tools like, LEX.	PSO -2,5 CO-1,2,3,4 PO-1,2 PI 1.6.1,2.5.2,2.5.3,3.7.1,5.4.1,5.4.2
3	**SYNTAX ANALYSIS**	Define CFG's. Ambiguity in CFG, Associativity and Precedence. Parsing- Top down parsing, Recursive descent parsing, predictive parsing, Bottom up parsing, Operator precedence grammars and parsing, LR parsers and their types (SLR, Canonical, LALR), YACC (Yet Another Compiler Compiler).	PSO -2,5 CO-1,2,3,4 PO-1,2 PI 1.6.1, 2.5.2,2.5.3,3.7.1,5.4.1,5.4.2
4	**SYNTAX DIRECTED TRANSLATION**	Types of attributes: Inherited and synthesized attributes, Dependency graphs, Evaluation order, Bottom up evaluation of S- attributed definitions L- attributed definitions and top down translation of attributes.	PSO -2,5 CO-1,2,3 PO-1,2 PI 1.6.1,2.5.2,2.5.3,3.7.1
5	**TYPE CHECKING**	Type system, Type expressions, Structural and name equivalence of types, Type conversions.	PSO -2,5 CO-1,2,3 PO-1,2 PI 1.6.1,2.5.2,2.5.3,3.7.1
6	**RUN TIME ENVIRONMENTS**	Storage organization, Storage-allocation strategies, Access to nonlocal names, Activation tree, Activation records, Parameter passing, Symbol table and dynamic storage allocations.	PSO -2,5 CO-1,2,3 PO-1,2 PI 1.6.1,2.5.2,2.5.3,3.7.1
7	**INTERMEDIATE**	Intermediate representations, Translation of declarations,	PSO -2,5

SL. No.	Module	Learning Objective	Program Specific Outcomes(PSO)/ Course Outcomes(CO)/ Program Outcomes(PO)/PI
	CODE GENERATION	Assignments, Control flow, Boolean expressions and procedure calls.	CO-1,2,3 PO-1,2 PI 1.6.1,2.5.2,2.5.3,3.7.1
8	**CODE GENERATION**	Issues in the design of a code generator, Basic blocks and flow graphs, Next use information, Register allocation, Code generation algorithm, DAG representation of programs, Code generation from DAGs, Peephole optimization and code generators.	PSO -2,5 CO-1,2,3 PO-1,2 PI 1.6.1,2.5.2, 2.5.3,3.7.1

Subject In-Charge **Subject Coordinator** **HOD/CSE**

SYLLABUS PLANNING AND COURSE COVERAGE REPORT

Date of Commencement of Semester:
Last date for completing the syllabus:
Total No. of Remedial classes held:
Any extra classes required to complete the syllabus: Yes/No

Module/ Chapter No.	Lecture No.	Topics/Experiments/Program Planned	Alloted hrs.	Date / Week No.	Covered (Yes/No)	Reason if not covered	CO Number Covered	Learning Resources used	Signature of Teacher	Signature of CR	Signature of DAC / HOD
1	1	Discuss the concepts of language processors: compiler & interpreter.									
		What is the difference between a compiler and interpreter?									
		What are the advantages of a compiler over an interpreter?									
	2	Analysis and Synthesis phase of a compiler.									
		What is a machine language & assembly language?									
	3	How a program, using C compiler, is executed on a host machine?									
		Given an assignment statement, show the translation process in different phases of a compiler.									
	4	Explain the different compiler construction tools.									
		What is a java language processor?									

Module/ Chapter No.	Lecture No.	Topics/Experiments/Program Planned	Alloted hrs.	Date / Week No.	Covered (Yes/No)	Reason if not covered	CO Number Covered	Learning Resources used	Signature of Teacher	Signature of CR	Signature of DAC / HOD
2	5	What are the roles of lexical analyzer?									
		How lexical can be divided into phases? Explain with an example									
	6	What are token, lexeme and pattern?									
		Find token, lexeme for a given statement.									
	7	Explain the concept of input buffering scheme in lexical analyzer.									
		Define sentinels. Given an example.									
	8	What is regular expression, Regular definition? How to draw transition for an expression?									
		Draw the transition diagram of C identifier.									
3	9	What are the roles of Parser? What is Context Free Grammar?									
		What is an ambiguity in CFG? Give an example.									
	10	What is left recursion and left factoring? How to remove left recursion with example.									
		Why is left recursion a problem in syntax analysis?									

Module/ Chapter No.	Lecture No.	Topics/Experiments/Program Planned	Alloted hrs.	Date / Week No.	Covered (Yes/No)	Reason if not covered	CO Number Covered	Learning Resources used	Signature of Teacher	Signature of CR	Signature of DAC / HOD
	11	What are the different types of parsing technique? Explain Top down parsing Technique-Recursive descent?									
		Perform recursive descent parsing techniques for a given grammar: Check whether a given string is accepted by the above grammar or not.									
	12	What is First set? What are the rules of generating it?									
		Give the first set of a given grammar:									
	13	What is a follow set? What are its rules? Explain with example.									
		Give the follow set of a grammar.									
	14	What is LL (1) grammar? What are rules of constructing predictive parsing table?									
		Create the parsing table for a given production rule.									
	15	Explain non-recursive predicting parsing with suitable example.									
		Perform non-recursive predicting parsing for a given grammar:									
	16	How to recover error in predictive parsing? What are the bottoms up parsing									

Module/ Chapter No.	Lecture No.	Topics/Experiments/Program Planned	Alloted hrs.	Date / Week No.	Covered (Yes/No)	Reason if not covered	CO Number Covered	Learning Resources used	Signature of Teacher	Signature of CR	Signature of DAC / HOD
		techniques?									
		Construct a parse tree from the leaves to the root using rightmost derivation in reverse from a given grammar and a string.									
	17	What is a Shift Reduce Parsing (SRP) technique? Explain with suitable example. Conflict of SRP.									
		Check whether a given string is parsed by the SRP									
	18	What is an operator grammar? Explain the precedence parsing algorithm with suitable example.									
		Check whether a given string can be parsed by **Operator-precedence Parser** and also show the stack implementation.									
	19	What are conflicts that arise in Bottom-up-parsing? How to recover from error in predictive parsing?									
		Construct the precedence function for the grammar with a given example.									
	20	What is a LR parser? How to construct canonical LR item sets for a given grammar.									

Module/ Chapter No.	Lecture No.	Topics/Experiments/Program Planned	Alloted hrs.	Date / Week No.	Covered (Yes/No)	Reason if not covered	CO Number Covered	Learning Resources used	Signature of Teacher	Signature of CR	Signature of DAC / HOD
		What is the difference between LR(0) and LL(1) item set?									
4	21	What is semantic analyzer? What is Syntax-Directed Definition (SDD)? Provide a suitable example. Explain Attribute Grammars.									
		Give the role of semantic analyzers? Which tool is used to generate it									
	22	Differentiate between synthesized and inherited attributes. Provide a Syntax-directed definition for a simple desk calculator.									
		What is the difference between syntax tree and parse tree?									
	23	What is annotated parse tree? Explain with an example.									
		Design an annotated parse tree for the expression 4*5.									
	24	Brief concept about translation scheme and dependency graph with example. How to perform evaluation order for SDD using a dependency graph?									
		Draw a dependency graph for the following expression:									

Module/ Chapter No.	Lecture No.	Topics/Experiments/Program Planned	Alloted hrs.	Date / Week No.	Covered (Yes/No)	Reason if not covered	CO Number Covered	Learning Resources used	Signature of Teacher	Signature of CR	Signature of DAC / HOD
	25	What is an abstract syntax tree? Write down the steps for creating a syntax tree.									
		Create a syntax tree for a given expression.									
	26	Define S-attributes & L-attributes? How bottom-up evaluation is performed for S-attributes?									
		Show the bottom-up evaluation of a given expression.									
5	27	Define a type system. Why do we need type system? What is Type Checking? What is its specification?									
		State the uses of Type checking									
	28	What are Type expressions? Explain equivalence of type expression. What is unification algorithm?									
		How to perform unification of two nodes?									
	29	Discuss the storage organization in compiler.									
		Differentiate between static and dynamic data storage allocations.									
	30	Explain the activation record structure.									

Module/ Chapter No.	Lecture No.	Topics/Experiments/Program Planned	Alloted hrs.	Date / Week No.	Covered (Yes/No)	Reason if not covered	CO Number Covered	Learning Resources used	Signature of Teacher	Signature of CR	Signature of DAC / HOD
6		What is an activation tree? Explain with suitable examples									
	31	What is intermediate code generator? What are its different types of representation?									
		Give the postfix notation for a given expression.									
	32	What is a Three-Address Code? Describe the format of a TAC? What are the types of three-address statements?									
		For the following assignment statement, provide the TAC representation for a given expression.									
	33	How to generate S-attribute definition for an assignment statement and while statement?									
		Draw the syntax tree for									
	34	What is symbol table? What is its type? Explain how an array is represented in TAC.									
		Problems based on TAC.									
7	35	Explain the concept of control flow: Short-circuit code & Boolean Expressions									

Module/ Chapter No.	Lecture No.	Topics/Experiments/Program Planned	Alloted hrs.	Date / Week No.	Covered (Yes/No)	Reason if not covered	CO Number Covered	Learning Resources used	Signature of Teacher	Signature of CR	Signature of DAC / HOD
		For the given statement show the representation of jumping code and control flow translations.									
	36	Show the SDD to produce TAC for Booleans. Discuss the concept of back patching with suitable example.									
		Give the SDD and also draw the annotated tree for a given expression.									
8	37	Consider the program fragment for dot operator and compute its TAC. Define basic block with suitable example.									
		Discuss the issues faced in code generation.									
	38	Discuss the various transformations on basic blocks. What is a flow graph?									
		Show how to represent the flow of a graph for a pseudo code.									
	39	Explain the simple code generator. Give the TAC and code generator for a given expression.									
		Generate the code for a statement.									
		What is Directed Acyclic Graph (DAG)? Explain the steps to construct it.									
		Construct the DAG for a									

Module/ Chapter No.	Lecture No.	Topics/Experiments/Program Planned	Alloted hrs.	Date / Week No.	Covered (Yes/No)	Reason if not covered	CO Number Covered	Learning Resources used	Signature of Teacher	Signature of CR	Signature of DAC / HOD
		given mathematical expression.									
8	40	Explanation of basic blocks and their methods of optimization. Introduction to Code generation.									

Subject In-Charge **Subject Coordinator** **HOD/CSE**

REMEDIAL CLASSES

DATE	NO OF STUDENTS ATTENDED	TOPICS COVERED	REMARKS

Detailed discussions (if any):

ATTENDANCE RECORDS

Sl	Name																																													

