MANIPAL INSTITUTE OF TECHNOLOGY
MANIPAL
(A constituent unit of MAHE, Manipal)

COURSE PLAN

Department	**:**	Computer Science and Engineering
Course Name & code	**:**	Compiler Design & CSE 3201
Semester & branch	**:**	VI & CSE
Name of the faculty	**:**	Mr. Shyam Karanth
No of contact hours/week:		02 01 00 03

ASSESSMENT PLAN

Course Outcomes (COs)

	At the end of this course, the student should be able to:	**No. of Contact Hours**	**Marks**
CO1:	Develop a familiarity on different phases of a compiler and recognize steps involved in lexical analyser generators.	7	19
CO2:	Describe top down and bottom up parsing techniques.	9	26
CO3:	Identify ambiguous grammars and analyse syntax directed translation techniques.	5	14
CO4:	Translate expressions into three address code and elaborate code generation phase.	12	33
CO5:	Discuss storage organization issues and make use of LEX and YACC tools.	3	8
	Total	36	100

Components	Surprise Quizzes	Sessional Tests	End Semester/ Make-up Examination
Duration	20 to 30 minutes	60 minutes	180 minutes
Weightage	20 % (4 X 5 marks)	30 % (2 X 15 Marks)	50 % (1 X 50 Marks)
Typology of Questions	Understanding/ Comprehension; Application; Analysis; Synthesis; Evaluation	Knowledge/ Recall; Understanding/ Comprehension; Application	Understanding/ Comprehension; Application; Analysis; Synthesis; Evaluation
Pattern	Answer one randomly selected question from the problem sheet (Students can refer their class notes)	MCQ: 10 questions (0.5 marks) Short Answers: 5 questions (2 marks)	Answer all 5 full questions of 10 marks each. Each question may have 2 to 3 parts of 3/4/5/6/7 marks
Schedule	4, 7, 10, and 13th week of academic calendar	Calendared activity	Calendared activity
Topics Covered	Quiz 1 (L 1-9 & T 1-9) **(CO 1)**	Test 1 (L 1-16 & T 1-16) **(CO 1&2)**	Comprehensive examination covering full syllabus. Students are expected to answer all questions **(CO 1-5)**
	Quiz 2 (L **10-18** & T 10-18) **(CO 2 &3)**		
	Quiz 3 (L 19-27 & T 19-27) **(CO 3 &4)**	Test 2 (L 17-32 & T 17-32) **(CO 3&4)**	
	Quiz 4 (L 28-36 & T 28-36) **(CO 5)**		

Course Plan

L. No./ T. No.	Topics	Course Outcome Addressed
L0	Introduction to the course	CO1
L1	Language Processors, The Structure of a Compiler- Lexical Analysis, Syntax Analysis, Semantic Analysis	CO1
L2	Intermediate Code Generation, Code Optimization, Code Generation, Symbol-Table Management	CO1
L3	The Role of the Lexical Analyzer, Input Buffering	CO1
L4	Recognition of Tokens, Architecture of a Transition-Diagram-Based Lexical Analyzer	CO1
L5	Tutorial on Recognition of Tokens	CO1
L6	Design of a Lexical Analyzer Generator- The Structure of the Generated Analyzer, Pattern Matching Based on NFAs	CO1
L7	Tutorial on Pattern Matching Based on NFAs	CO1
L8	Syntax Analysis - Introduction, Writing a Grammar- Lexical versus Syntactic Analysis, Eliminating Ambiguity and Left Recursion, Left Factoring	CO2
L9	Tutorial on Eliminating Ambiguity and Left Recursion	CO2

L10	Top-Down Parsing - Recursive-Descent, First and Follow, LL(1) Grammars, Nonrecursive Predictive Parsing	CO2
L11	Error Recovery in Predictive Parsing, Bottom-Up Parsing - Reductions, Handle Pruning	CO2
L12	Tutorial on Predictive Parsing	CO2
L13	Shift-Reduce Parsing, Introduction to LR parsing- Simple LR, Why LR Parsers?	CO2
L14	Items and LR(0) Automaton, The LR-Parsing Algorithm, Constructing SLR-Parsing Tables	CO2
L15	Tutorial on LR parsing	CO2
L16	More Powerful LR parsers- Canonical LR(1) Items, Constructing LR(1) Sets of Items, Canonical LR(1) Parsing Tables	CO2
L17	Using Ambiguous Grammars-Precedence and Associativity to Resolve Conflicts	CO3
L18	Tutorial on Ambiguous Grammars	CO3
L19	Syntax-Directed Translation - Syntax-Directed Definitions, Evaluation Order for SDD's- Dependency Graphs	CO3
L20	Ordering the Evaluation of Attributes, Applications of Syntax-Directed Translation - Construction of Syntax Trees	CO3
L21	Tutorial on Construction of Syntax Trees	CO3
L22	Intermediate-Code Generation - Variants of Syntax Trees	CO4
L23	Three Address Code- Addresses and Instructions, Quadruples, Triples	CO4
L24	Tutorial on Three Address Code	CO4
L25	Types and Declarations- Type Expressions, Type Equivalence, Declarations	CO4
L26	Translation of Expressions- Operations Within Expressions	CO4
L27	Tutorial on Translation of Operations within expressions	CO4
L28	Code Generation - Issues in Design of Code Generator	CO4
L29	The Target Language, Basic Blocks and Flow Graphs	CO4
L30	Optimization of Basic Blocks- The DAG Representation of Basic Blocks	CO4
L31	Tutorial on Optimization of Basic Blocks	CO4
L32	Peephole Optimization, Register Allocation and Assignment- Global Register Allocation, Run-Time Environments - Storage Organization	CO4
L33	Tutorial on Peephole Optimization	CO4
L34	Stack Allocation of Space- Activation Trees, Activation Records	CO5
L35	Theory of FLEX- Structure of a FLEX program, Regular Expression, FLEX library functions	CO5
L36	Theory of YACC- YACC Symbols, Symbol values, Symbol Types, YACC Library	CO5

References:

1. Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman, “Compilers Principles, Techniques and Tools”, Pearson Education, 2nd edition. 2010.
2. Vinu V. Das, “Compiler Design using FLEX and YACC”, Prentice-Hall, 2007.
3. Kenneth C. Louden, “Compiler Construction - Principles and Practice”, Thomson, First Edition, 2007.
4. John R. Levine, Tony Manson, Doug Brown, “LEX & YACC”, O Reilly Media, Second Edition, 2012.
5. Click or tap here to enter text.
6. Click or tap here to enter text.
7. Click or tap here to enter text.

Submitted by: Shyam Karanth

(Signature of the faculty)

Date: 12-01-2019

Approved by: Dr. Ashalatha Nayak

(Signature of HOD)

Date: 12-01-2019

FACULTY MEMBERS TEACHING THE COURSE (IF MULTIPLE SECTIONS EXIST):

FACULTY	SECTION	FACULTY	SECTION
Mr. Shyam Karanth	A		
Ms. Priya Kamath B	B		
Ms. Deepthi S	C		
Ms. Roopashri	D		
