

## NIRMA UNIVERSITY

<b>Institute:</b>	Institute of Technology
<b>Name of Programme:</b>	Master of Computer Application (2-Years Programme)
<b>Course Code:</b>	3MCAD308
<b>Course Title:</b>	Compiler Construction
<b>Course Type:</b>	Departmental Elective
<b>Year of Introduction:</b>	2021-22

### Credit Scheme

L	T	Practical Component				C
		LPW	PW	W	S	
3	0	0	-	-	-	3

### Course Learning Outcomes (CLO):

At the end of the course, students will be able to –

1. summarize the functionalities of various phases of compiler
2. apply language theory concepts to various phases of compiler design
3. identify appropriate optimization technique for compilation process
4. design a miniature compiler using appropriate compiler design concepts

### Syllabus:

**Total Teaching hours: 45**

Unit	Syllabus	Teaching hours
Unit-I	<b>Introduction to Compiling:</b> Compilers, Analysis of the Source Program, Phases of Compiler.	02
Unit-II	<b>Lexical Analysis:</b> The Role of Lexical Analyzer, Input Buffering, Specification and Recognition of Tokens, Finite Automata, NFA and DFA Construction, NFA to DFA Conversion, Optimization of DFA based Pattern Matcher.	09
Unit-III	<b>Syntax Analysis and Syntax Directed Translations:</b> The Role of Parser, Context Free Grammar, Writing Grammar, Top-Down and Bottom-Up Parsing, Operator Precedence Parsing, LR Parsers, Construction of Syntax Trees, Bottom Up Evaluation of Attributes.	14
Unit-IV	<b>Type Checking:</b> Type Systems, Specification of Simple Type Checkers, Equivalence of Type Expressions, Type Conversions.	03
Unit- V	<b>Run-Time Environments:</b> Source Language Issues, Storage Organizations and Allocation Strategies, Parameter Passing, Symbol Table, Introduction to Dynamic Storage Allocation.	06
Unit-VI	<b>Intermediate Code Generation:</b> Intermediate Languages, Declarations, Assignment Statements, Boolean Statements, Back-patching and Procedure Calls.	04
Unit-VII	<b>Code Generation and Optimization:</b> Issues in the Design of Code Generator, Target Machines, Run-time Storage Management, Basic Blocks and Flow Graphs, Basics of Code Optimization, Optimization of Basic Blocks, Loops in Flow Graph, Introduction to Global Data Flow Analysis and Code Improving Transformations.	07

Self-Study: The self-study contents will be declared at the commencement of semester. Around 10% of the questions will be asked from self-study contents.

- Suggested Readings/  
References:
1. A.V. Aho, Ravi Sethi and J.D.Ullman, Compilers: Principles, Techniques and Tools, Addison Wesley
  2. D. M. Dhamdhare, Compiler Construction-Principles and Practices, Mcmillian
  3. Trembley J.P. And Sorenson, The Theory and Practice of Compiler Writing, Tata McGraw Hill
  4. Waite W.N. And Goos G, Compiler Construction, Springer Verlag
  5. Sudha Sadasivam, Compiler Design, Scitech Publications Pvt Ltd
  6. Das, Compiler Design Using Flex and Yacc, PHI

Suggested List of Experiments: -NA-

Suggested Case List: -NA-