

NIRMA UNIVERSITY

Institute:	Institute of Technology
Name of Programme:	B.Tech. Computer Science and Engineering
Course Code:	2CS701
Course Title:	Compiler Construction
Course Type:	Core
Year of Introduction:	2021-22

Credit Scheme

L	T	Practical Component			C
		LPW	PW	W S	
3	0	2	-	-	4

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. develop a miniature compiler using appropriate compiler design tool

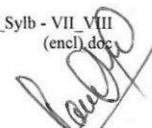
Syllabus:

Total Teaching hours: 45

Unit	Syllabus	Teaching hours
Unit-I	Introduction: Overview of the Translation Process, Structure of a compiler, Types of compiler and applications, Symbol table	03
Unit-II	Lexical Analysis: The role of a Lexical Analyzer, Input Buffering, Specifications of Tokens, Recognition of tokens, Lexical Analyzer Generator, Finite Automata, Regular Expression to Automata, Optimization of DFA based Pattern Matching	06
Unit-III	Syntax Analysis: Context Free Grammar, Top-down Parsing, Bottom-up Parsing, LR Parsers, Error Recovery, Parsing for ambiguous grammars, Parsing Generator Tools	13
Unit-IV	Syntax Directed Translation: Syntax Directed Definition (SDD), Evaluation order of SDD, Syntax Directed Translation Schemes	07
Unit-V	Intermediate Code Generation: Variants of Syntax Trees, Three Address Codes, Type Checking, Control Flow, Back patching	07
Unit-VI	Runtime Environment: Storage Organization, Stack Allocation and Heap Management	02
Unit VII	Code Generation and Optimization: Issues in code generation, Data Flow and Control Flow, Peephole Optimization, Register Allocation, Machine independent optimization techniques	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. Aho, Lam, Ullman, Sethi, Compilers, Principles, Techniques and Tools, Pearson
2. Keith D Cooper & Linda Torczon, Engineering a Compiler, Elsevier
3. Jean Paul Trembly & Paul G Sorenson, The theory and Practice of Compiler writing, McGraw Hill

Suggested List of Experiments:

Sr. No.	Practical Title	Hours
1	To implement lexical analyse to recognize all distinct token classes.	02
2	To implement a Recursive Descent Parser Algorithm for the grammar.	02
3	To find the First () and Follow () of a grammar.	04
4	To implement the Left most derivation removal algorithm.	04
5	To implement a calculator in YACC.	04
6	To generate Three Address code for assignment statement	02
7	To implement grammar rules for control statements, and Loop control.	04
8	To implement a Type Checker.	02
9	To implement Assembly code generator.	02
10	To implement Code Optimization techniques.	04

Suggested Case List:

-NA-