BCSE307L	Compiler Design		L	Т	Р	С
			3	0	0	3
Pre-requisite	NIL	Syllabus version				
		1.0				

#### **Course Objectives**

- 1. To provide fundamental knowledge of various language translators.
- 2. To make students familiar with lexical analysis and parsing techniques.
- 3. To understand the various actions carried out in semantic analysis.
- 4. To make the students get familiar with how the intermediate code is generated.
- 5. To understand the principles of code optimization techniques and code generation.
- 6. To provide foundation for study of high-performance compiler design.

#### **Course Outcomes**

- 1. Apply the skills on devising, selecting, and using tools and techniques towards compiler design
- 2. Develop language specifications using context free grammars (CFG).
- 3. Apply the ideas, the techniques, and the knowledge acquired for the purpose of developing software systems.
- 4. Constructing symbol tables and generating intermediate code.
- 5. Obtain insights on compiler optimization and code generation.

#### Module:1 INTRODUCTION TO COMPILATION AND LEXICAL ANALYSIS 7 hours

Introduction to LLVM - Structure and Phases of a Compiler-Design Issues-Patterns-Lexemes-Tokens-Attributes-Specification of Tokens-Extended Regular Expression- Regular expression to Deterministic Finite Automata (Direct method) - Lex - A Lexical Analyzer Generator.

#### Module:2 | SYNTAX ANALYSIS

8 hours

Role of Parser- Parse Tree - Elimination of Ambiguity - Top Down Parsing - Recursive Descent Parsing - LL (1) Grammars - Shift Reduce Parsers- Operator Precedence Parsing - LR Parsers, Construction of SLR Parser Tables and Parsing- CLR Parsing- LALR Parsing.

## Module:3 | SEMANTICS ANALYSIS

5 hours

Syntax Directed Definition – Evaluation Order - Applications of Syntax Directed Translation - Syntax Directed Translation Schemes - Implementation of L-attributed Syntax Directed Definition.

## Module:4 | INTERMEDIATE CODE GENERATION

5 hours

Variants of Syntax trees - Three Address Code- Types - Declarations - Procedures - Assignment Statements - Translation of Expressions - Control Flow - Back Patching- Switch Case Statements.

#### Module:5 | CODE OPTIMIZATION

6 hours

Loop optimizations- Principal Sources of Optimization -Introduction to Data Flow Analysis - Basic Blocks - Optimization of Basic Blocks - Peephole Optimization- The DAG Representation of Basic Blocks -Loops in Flow Graphs - Machine Independent Optimization-Implementation of a naïve code generator for a virtual Machine- Security checking of virtual machine code.

# Module:6 | CODE GENERATION

5 hours

Issues in the design of a code generator- Target Machine- Next-Use Information - Register Allocation and Assignment- Runtime Organization- Activation Records.

## Module:7 PARALLELISM

7 hours

Parallelization- Automatic Parallelization- Optimizations for Cache Locality and Vectorization- Domain Specific Languages-Compilation- Instruction Scheduling and Software Pipelining- Impact of Language Design and Architecture Evolution on Compilers-Static Single Assignment

## Module:8 | Contemporary Issues

2 hours

				Total L	ecture hours:	45 hours			
Text Book(s)									
1.	. A. V. Aho, Monica S. Lam, Ravi Sethi and Jeffrey D. Ullman, Compilers: Principles,								
	techniques, & tools, 2007, Second Edition, Pearson Education, Boston.								
Reference Books									
1.	Watso	son, Des. A Practical Approach to Compiler Construction. Germany, Springer							
	Interna	national Publishing, 2017.							
Mode of Evaluation: CAT, Quiz, Written assignment and FAT									
Recommended by Board of Studies 04-03-2022									
App	proved b	y Academic Council	No. 65	Date	17-03-2022				