Course Code	Course name	L	T	P	C
CSEG3015	**Compiler Design**	3	0	0	3
Total Units to be Covered: 5	**Total Contact Hours: 45**				
Prerequisite(s):	Automata Theory and Formal Languages Data structures, knowledge of automata theory, basic knowledge of computer architecture	**Syllabus version: 1.0**			

Course Objectives

1. To introduce the major concept areas of language translation and compiler design.
2. To enrich the knowledge in various phases of compiler and its use, code optimization techniques, machine code generation, and use of symbol table.
3. To extend the knowledge of parser by parsing LL parser and LR parser.
4. To provide practical programming skills necessary for constructing a compiler.

Course Outcomes

On completion of this course, the students will be able to

CO 1. Comprehend different phases of compiler.

CO 2. Use concepts of regular grammar to build lexical analyzer.

CO 3. Build parsers for a context free grammar.

CO 4. Synthesize syntax directed translations rules.

CO 5. Assess code and memory optimization techniques to improve the performance of a program.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	2	2	3	-		-	-	-	-	-	-	-	2	-	-
CO 2	2	2	3	-	1	-	-	-	-	-	-	-	2	-	-
CO 3	2	2	3	-	1	-	-	-	-	-	-	-	2	-	-
CO4	2	2	3	-	1	-	-	-	-	-	-	-	2	-	-
CO5	2	2	3	-	-	-	-	-	-	-	-	-	2	-	-

Average	2	2	3	-	.6	-	-	-	-	-	-	-	2	-	-

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

Syllabus

Unit I: Introduction **10 Lecture Hours**

Compiler, Phases and Passes, Bootstrapping, Finite State Machines and Regular Expressions and their Applications to Lexical Analysis, Implementation of Lexical Analyzers, Lexical Analyzer Generator, LEX, Formal Grammars and their Applications to Syntax Analysis, BNF Notation, Ambiguity, YACC. The Syntactic Specification of Programming Languages: Context Free Grammars, Derivation and Parse Tree, Capabilities of CFG.

Unit II: Basic Parsing Techniques **10 Lecture Hours**

Parsers, Shift Reduce Parsing, Operator Precedence Parsing, Top Down Parsing, Predictive Parsing, Automatic Construction of Efficient Parsers: LR Parsers, The Canonical Collection of LR(0) items, Constructing SLR Parsing Tables, Constructing Canonical LR Parsing Tables, Constructing LALR Parsing Tables, Using Ambiguous Grammars, An Automatic Parser Generator, Implementation of LR Parsing Tables, Constructing LALR set of items

Unit III: Syntax-Directed Translation **10 Lecture Hours**

Syntax Directed Translation Schemes, Implementation of Syntax Directed Translators, Intermediate Code, Postfix Notation, Parse Tree & Syntax Tree, Three Address Code, Quadruples & Triples, Translation of Assignment Statements, Boolean Expressions, Statements that alters the Flow of Control, Postfix Translation, Translation with a Top Down Parser, More about Translation: Array Reference in Arithmetic Expressions, Procedure Calls, Declaration, and Case Statements.

Unit IV: Symbol Table **10 Lecture Hours**

Data Structures for Symbol Tables, Representing Score Information, Run Time Administration: Implementation of Simple Stack Allocation Scheme, Storage Allocation in Block Structures Language, Error Detection and Recovery: Lexical Phase Error, Syntactic Phase Errors, Semantic Phase Errors.

Unit V: Introduction to Code Optimization **5 Lecture Hours**

Loop Optimization, the DAG Representation of Basic Blocks, Value Number and Algebraic Laws, Global Data-Flow Analysis

Total lecture Hours 45

Textbooks

1. Alfred V. Aho, Ravi Sethi, and Jeffrey D. Ullman, "Compilers- Principles, Techniques, and Tools", 2nd Edition, Pearson Education Asia, 2013.
2. Robin Hunter, "The Essence of Compiler", 2nd Edition, Pearson Publication, 2004.

Reference Books

1. Randy Allen, and Ken Kennedy, "Optimizing Compilers for Modern Architectures: A Dependence-based Approach", Morgan Kaufmann Publishers, 2002.
2. Steven S. Muchnick, "Advanced Compiler Design and Implementation", Morgan Kaufmann Publishers - Elsevier Science, India, Indian Reprint 2003.
3. Keith D Cooper, and Linda Torczon, "Engineering a Compiler", Morgan Kaufmann Publishers Elsevier Science, 2004.
4. Charles N. Fischer, and Richard. J. LeBlanc, "Crafting a Compiler with C", Pearson Education, 2008.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	IA	MID SEM	End Sem	Total
Weightage (%)	50	20	30	100

Detailed breakup of Internal Assessment

Internal Assessment Component	Weightage in calculation of Internal Assessment (100 marks)
Quiz 1	15%
Quiz 2	15%
Class Test 1	15%
Class Test 2	15%
Assignment 1/Project	20%
Assignment 2/Project	20%