Name of Department:- Computer Science and Engineering

1.	Subject Code:	TCS 601	Course Title:	Compiler Design	
2.	Contact Hours:	L: 3	T: -	P: -	

3. Semester: VI

4. Pre-requisite: TCS 501

5. Course Outcomes: After completion of the course students will be able to

1. Understand the various phases and fundamental principles of compiler design like lexical, syntactical, semantic analysis, code generation and optimization.
2. Compare and contrast various parsing techniques such as SLR, CLR, LALR etc.
3. Use annotated tree to design the semantic rules for different aspects of programming language.
4. Implement lexical analyzer and parser by using modern tools like Flex and Bison.
5. Examine patterns, tokens & regular expressions for solving a problem in the field of data mining.
6. Design a compiler for concise programming language.

6. Detailed Syllabus

UNIT	CONTENTS	Contact Hrs
Unit – I	**Introduction, Lexical analysis:** Compilers; Analysis of Source Program; The Phases of a Compiler; Cousins of the Compiler; The grouping of phases; Compiler- Construction tools. Lexical analysis: The Role of Lexical Analyzer; Input Buffering; Specifications of Tokens; Recognition of Tokens.	**9**
Unit - II	**Syntax Analysis – 1:** The Role of the Parser; Context-free Grammars; Writing a Grammar; Top-down Parsing; Bottom-up Parsing. Operator-Precedence Parsing; LR Parsers; Using ambiguous grammars; Parser Generators	**9**
Unit – III	**Syntax-Directed Translation:** Syntax-Directed definitions; Constructions of Syntax Trees; Bottom-up evaluation of S-attributed definitions; L-attributed definitions; Top-down translation. **Run-Time Environments :** Source Language Issues; Storage Organization; Storage-allocation strategies, Storage-allocation in C; Parameter passing	**8**
Unit – IV	**Intermediate Code Generation:** Intermediate Languages; Declarations; Assignment statements; Boolean Expressions; Case statements; Back patching; Procedure calls. **Code Generation:** Issues in the design of Code Generator; The Target Machine; Run-time Storage Management; Basic blocks and Flow graphs; Next-use information; A Simple Code Generator; Register allocation and assignment; The dag representation of basic blocks; Generating code from dags.	**9**

Unit – V	**Code Optimization, Compiler Development:** Code Optimization: Introduction; The principal sources of optimization; Peephole optimization; Optimization of basic blocks; Loops in flow graphs. Compiler Development: Planning a compiler; Approaches to compiler development; the compiler development environment; Testing and maintenance.	**9**
	Total	**44**

Text Books:

1. Alfred V Aho, Ravi Sethi, Jeffrey D Ullman: "Compilers- Principles, Techniques and Tools", Pearson Education, 2007.

Reference Books:

1. Charles N. Fischer, Richard J. leBlanc, Jr.:" Crafting a Compiler with C", Pearson Education, 1991.
2. Andrew W Apple: "Modern Compiler Implementation in C", Cambridge University Press, 1997.
3. Kenneth C Louden: "Compiler Construction Principles & Practice", Thomson Education, 1997.