Name of Department: - Computer Science and Engineering

1. Subject Code: TCS 601 Course Title:
2. Contact Hours: L: 3 _ T: _ P:
3. Semester: VI
4. Pre-requisite: TCS-402
5. Course Outcomes: After completion of the course students will be ableto

1. Appraise the 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 a concise programming language.

6. Detailed Syllabus

UNIT	CONTENTS	CONTACT Hrs
Unit – I	**Introduction:** Compiler Introduction; Analysis of source program; Phases and Passes of Compiler; Symbol table & its implementation; Cousins of a Compiler; Compiler Construction Tools; Bootstrapping: Regular Grammar and Regular Expressions. **Lexical analysis:** Role of a Lexical Analyzer; Input Buffering; Specifications of Tokens; Recognition of Tokens; LEX Tool and its Implementation	9
Unit - II	**Syntax Analysis:** Introduction toCFG; Writing a Grammar; Ambiguous Grammars; Role of a Parser; Basic Parsing Techniques; Top-down Parsing; Bottom-up Parsing; Operator-Precedence Parsing; Parser Generators (YACC)	10
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-Allocation Strategies, Parameter Passing: Stack/Heap Allocation. Error Handling	10
Unit – IV	**Intermediate Code Generation (ICG):** Intermediate Code; ICG using Postfix Notation, Syntax Tree, Directed Acyclic Graph (DAG); Three Address Code; Quadruples & Triples; Back Patching; Intermediate Languages; Declarations; Assignment Statements; Boolean Expressions; Case Statements;; Procedure Calls; Array References: **Code Optimization:** Introduction toCode Optimization; Principal Sources of Optimization; Machine Dependent & Independent Code Optimization; Peephole optimization; Global and Local Optimization of Basic Blocks.	12
Unit – V	**Code Generation:** Code Generation Issues; The Target Machine; Basic Blocks And Flow Graphs; Next-Use Information; A Simple Code Generator; Register Allocation & Assignment; DAG Representation of Basic Blocks; Generating CodeFrom DAG. **Compiler Development**: Planning a Compiler; Compiler Development Approaches; Compiler development environment; Testing &Maintenance.	9
	Total	50

Text Books:

1. AlfredVAho,RaviSethi,JeffreyDUllman:"Compilers-Principles,TechniquesandTools", 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.