Course Title: Complier Design and Construction
Course no: CSC-352 **Full Marks**: 70+10+20
Credit hours: 3 **Pass Marks**: 28+4+8

Nature of course: Theory (3 Hrs.) + Lab (3 Hrs.)

Course Synopsis: Analysis of source program. The phases of compiler.

Goal: This course introduces fundamental concept of compiler and its different phases.

Course Contents:

Unit. 1: **6 Hrs.**

1.5 Introduction to compiling: Compilers, Analysis of source program, the phases of compiler, compiler-construction tools.
1.6 A Simple One-Pass Compiler: Syntax Definition, Syntax directed translation, Parsing, Translator for simple expression, Symbol Table, Abstract Stack Machines.

Unit 2: **19 Hrs.**

2.4 Lexical Analysis: The role of the lexical analyzer, Input buffering, Specification of tokens, Recognition of tokens, Finite Automata, Regular Expression to an NFA, Design of a lexical analyzer generator 5 hrs.
2.5 Syntax Analysis: The role of parser, Context free grammars, Writing a grammars, Top-down parsing, Bottom-up parsing, Operator-preceding parsing, LR parsing, Ambiguous grammar.
2.6 Syntax Directed Translation: Syntax-directed definition, Syntax tree and its construction, Evaluation of S-attributed definitions, L-attributed, Top-down translation, Recursive evaluators.
2.7 Type Checking: Type systems, Specification of a simple type checker, Type conversions.

Unit 3: **13 Hrs.**

3.3 Intermediate Code Generation: Intermediate languages, Declarations, Assignments Statements, Boolean Expressions, Back patching.
3.4 Code Generator: Issues in design of a code generator, the target machine, Run – time storage management, Basic blocks and flow graphs, a simple code generator, Peephole organization, Generating code from dags, Dynamic programming code-generation algorithm, Code-generator generators.
3.5 Code Optimization: The principal sources of optimization, Optimization of basic blocks, loops in flow graphs.

Unit 4: **7 Hrs.**

4.6 Writing a Compilers: Planning a compiler, Approaches to compiler development, the compiler development environment, Testing and Maintenance.

4.7 Comparing some compliers: Pascal Complier, C compiler, C++ complier.

Laboratory works:

1 Writing a complier, optimization techniques, comparing the compilers.
2. Construction of Lexical Analyser.
3. Construction of Parser
4. Development of Code Generator
5. Write a code to show the function of symbol table.
6. Implement the parsing techniques.
7. Show the application of different types of grammar.
8. Implement the lexical analyzer generator.
9. Implement the type conversation.
10. The course instructor is allowed to create a group two students.
    a. Assign them to write a small compiler.

Text Books: Compilers, Principles, Techniques, and Tools, Pearson Education Asia.

Reference:

Homework Assignment: Assignment should be given from the above units in throughout the semester.

Computer Usage: No specific

Prerequisite: C, C++, Data Structure, Automata Theory

Category Content: Science Aspect: 25%
Design Aspect: 75%