**Tribhuvan University**

**Institute of Science and Technology**

**Bachelor of Science in Computer Science and Information Technology**

**Course Title: Compiler Design and Construction**

**Course Code: CSC376 | Semester: VI | Nature: Theory + Lab**

**Full Marks: 60 + 20 + 20 | Pass Marks: 24 + 8 + 8 | Credit Hours: 3**

## Course Overview

Course Description: This course is designed to develop acquaintance with fundamental concepts of compiler design. The course starts with the basic concepts and also includes different phases of compilers like lexical analysis, syntax analysis, syntax-directed translation, type checking, etc., in detail.

Course Objective:

- To develop knowledge in compiler design.
- To develop lexical analyzers, parsers, and small compilers using different tools.
- To develop lexical analyzers, parsers, and small compilers by using general-purpose programming languages.

## Unit 1: Introduction to Compiler (3 Hrs)

1.1 Compiler Structure: Analysis and Synthesis Model of Compilation, different sub-phases within analysis and synthesis phases.

1.2 Basic Compiler Concepts: Interpreter, simple One-Pass Compiler, preprocessor, macros, symbol table, and error handler.

## Unit 2: Lexical Analyzer (19 Hrs)

2.1 Lexical Analysis: Role of lexical analyzer, Input buffering, Specification and recognition of tokens, Finite Automata, Regular Expression to NFA, Design of a lexical analyzer generator.

2.2 Syntax Analysis: Role of parser, Context-free grammars, Writing grammars, Top-down parsing, Bottom-up parsing, Operator-preceding parsing, LR parsing, Ambiguous grammar.

2.3 Syntax Directed Translation: Syntax-directed definition, Syntax tree and its construction, Evaluation of S-attributed definitions, L-attributed, Top-down translation, Recursive evaluators.

2.4 Type Checking: Type systems, Specification of a simple type checker, Type conversions.

**Tribhuvan University**

**Institute of Science and Technology**

**Bachelor of Science in Computer Science and Information Technology**

**Course Title: Compiler Design and Construction**

**Course Code: CSC376 | Semester: VI | Nature: Theory + Lab**

**Full Marks: 60 + 20 + 20 | Pass Marks: 24 + 8 + 8 | Credit Hours: 3**

## Unit 3: Symbol Table Design and Runtime Storage Management (4 Hrs)

3.1 Symbol Table Design: Function of Symbol Table, Information provided, Attributes, and Data Structures for symbol table.

3.2 Runtime Storage Management.

## Unit 4: Intermediate Code Generation, Code Generation, Optimization, and Case Studies (19 Hrs)

4.1 Intermediate Code Generation: High and Low-level Intermediate representation, Syntax tree & DAG representations, Three-address code, Quadruples, Triples, SDT for intermediate code, Intermediate code generation for Declarations, Assignments, Control Flow, Boolean Expressions, and Procedure Calls; Back patching.

4.2 Code Generation: Factors affecting code generation, Target Language, Basic blocks, Flow graphs, Dynamic programming code-generation algorithm.

4.3 Code Optimization: Need and criteria of Code Optimization, Basic optimization techniques.

4.4 Case Studies of some compilers: C compiler, C++ compiler.

## Laboratory Work Guidelines

The laboratory work develops practical knowledge on different concepts of compiler design. Students should:

- Create a project using a lexical analyzer generator or any high-level language.
- Create a parser using a parser generator or any high-level language.
- Write programs for intermediate code generation and machine code generation.
- Create the front end of a compiler using general-purpose programming languages.

## Reference Books

1. Compilers: Principles, Techniques, and Tools by Alfred V. Aho, Ravi Sethi, Jeffrey D. Ullman; Pearson Education

2. Introduction to Automata Theory, Languages, and Computation by Johne E. Hopcroft, Rajeev Motwani, Jeffrey D.

Ulman; Pearson Education

3. Advanced Compiler Design and Implementation by Steven Muchnick; Morgan Kaufman Publication