# Capital University of Science and Technology

## Department of Computer Science

## CS4623 - Compiler Construction

| | |
|---|---|
| **Course Title:** | Compiler Construction (CS4623) |
| **Pre-requisite(s):** | Theory of Automata and Formal Languages (CS3613)<br>Computer Organization and Assembly Language (CS2523) |
| **Credit Hours:** | 3 |
| **Instructor(s):** | |
| **Text Book(s):** | Kenneth C. Louden: Compiler Construction – Principles and Practice, Course Technology, 1997, ISBN 978-0534939724 |
| **Reference Book(s):** | • Alfred V. Aho, Monica S. Lam, Ravi Sethi and Jeffrey D. Ullman: Compilers: Principles, Techniques and Tools, Addison-Wesley Publishing Company, 2006, ISBN 978-0321486813.<br>• D. Grune, H. Bal, C. Jacobs, K. Langendoen: Modern Compiler Design, Wiley, 2000, ISBN 978-0471976974. |
| **Web Reference:** | • https://www.tutorialspoint.com/compiler_design/ |

## Course Introduction:

This course presents a conceptual and practical introduction to imperative and object oriented programming, exemplified by C++. As well as providing grounding in the use of C++, the course will cover general principles of programming in imperative and object oriented frameworks. The course should enable you to develop programs that support experimentation, simulation and exploration in other parts of the Informatics curriculum (e.g. the capacity to implement, test and observe a particular algorithm).

## Course Objectives:

This course teaches the basic techniques used in compiler construction such as lexical analysis, top-down, bottom-up parsing, context-sensitive analysis, and intermediate code generation. It focuses on the basic data structures used in compiler construction such as abstract syntax trees, symbol tables, three-address code, and stack machines. We teach the students so that they can design and implement a compiler using a software engineering approach.

## Course Learning Outcomes (CLOs):

**CLO:1.** Describe the architecture of compiler, and formal notations to define a programming language. [C2-Understanding]

**CLO:2.** Implement syntax analyzer by using various top down and bottom up algorithms. [C3-Application]

**CLO:3.** Implement semantic analyzer through attribute grammars and symbol table.. [C3-Application]

**CLO:4.** Use appropriate code generation and optimization techniques. [C3-Application]

## CLOs – PLOs Mapping:

| | **CLO:1** | **CLO:2** | **CLO:3** | **CLO:4** |
|---|---|---|---|---|
| PLO:1 (Academic Education) | | | | |
| PLO:2 (Knowledge for Solving Computing Problems) | | | | |
| PLO:3 (Problem Analysis) | √ | | √ | |
| PLO:4 (Design/Development of Solutions) | | √ | | √ |
| PLO:5 (Modern Tool Usage) | | | | |

## Course Contents:

| **Week** | **Contents** |
|---|---|
| 1 | Overview of high-level languages and translation, Levels of programming languages, Compilation and Interpretation, Architecture of a compiler, Phases of the compilation process |
| 2 | Language definition, Syntax and semantic Specification<br>Using BNF notation to define syntax of a language |
| 3 | Chomsky's classification of grammars, Lexical analysis, Tokens and types of tokens<br>Regular Expressions and DFA’s |

| | |
|---|---|
| 4 | Implementing a Lexical Analyzer using DFA, Error handling in a Lexical Analyzer |
| 5 | LL(k) and LR(k) grammars, Syntax Analysis (Parsing), Parse Trees and Abstract Syntax Trees |
| 6 | Ambiguity and dangling else problem, Left-recursive grammars, Operator precedence, Left-factoring |
| 7 | Recursive Descent Parsing, LL(1) Parsing |
| 8 | Bottom Up Parsing- Shift Reduce Parsers, Simple Precedence Parsing |
| **Mid-Term Exam** | |
| 9 | Operator precedence parsing |
| 10 | DFA of LR(0) Items, SLR Parsing and conflicts |
| 11 | DFA of LR(1) Items, LALR Parsing |
| 12 | Static and dynamic semantic analysis, Symbol table, Type checking |
| 13 | Runtime environments and storage allocation, Runtime stack |
| 14 | Code generation, 3-address code, Code generation for expressions and assignments |
| 15 | Code generation for loops and control statements |
| 16 | Code Optimization techniques for front-end and back-end |

## Grading Policy:

| S.No | Grading | % of Total Marks |
|---|---|---|
| 1 | Assignments | 20 |
| 2 | Quizzes | 20 |
| 3 | Mid-term Exam | 20 |
| 4 | Final Exam | 40 |
| | **Total** | **100** |