Compiler Construction

Lecture # 01

Mr. Usman Wajid

usman.wajid@nu.edu.pk

January 24, 2023

Tentative Grade Distribution

Grades	Distribution
Quizes	05
Assignments	05
Class participation	05
Presentations	05
Final Project	10
1^{st} sessional	15
2^{nd} sessional	15
The final	40

Course Learning Outcomes (CLOs)

- Compiler construction techniques such as lexical analysis, Syntax Analysis and Intermediate code generation

Course Learning Outcomes (CLOs)

- Compiler construction techniques such as lexical analysis, Syntax Analysis and Intermediate code generation
- Basic data structures such as an abstract syntax trees, symbol tables, three-address code, and stack machines

Course Learning Outcomes (CLOs)

- Compiler construction techniques such as lexical analysis, Syntax Analysis and Intermediate code generation
- Basic data structures such as an abstract syntax trees, symbol tables, three-address code, and stack machines
- design and implement a compiler using a software engineering approach

Course Learning Outcomes (CLOs)

- Compiler construction techniques such as lexical analysis, Syntax Analysis and Intermediate code generation
- Basic data structures such as an abstract syntax trees, symbol tables, three-address code, and stack machines
- design and implement a compiler using a software engineering approach
- using generators (e.g. Lex and Yacc)

Referential books

Compilers: Principles, Techniques and Tools (2006)

(The Dragon book 2nd edition) A. V. Aho, R. Sethi, J. D. Ullman and M. S. Lam

Modern Compiler Design (2004)

D. Grune, H. E. Bal, C. J. H. Jacobs, K. G. Langendoen, John Wiley

Modern Compiler Implementation in C (2004)

A. W. Appel, M. Ginsburg, Cambridge University Press

Compiler Construction (2013)

K. V. N. Sunitha

Course Outlines

- Introduction
- Lexical Analyzer
- Syntax Definition — Grammers
- Syntax Analysis — Top-Down Parsers
- Bottom-Up Parsers
- Syntax Directed Translation
- Semantic Analysis
- Intermediate Code Generation
- Symbol Table
- Code Optimization
- Code Generation

Pre-requisites

- Theory of Automaton — DFA, NFA and CFG

Pre-requisites

- Theory of Automaton — DFA, NFA and CFG
- Data Structures — Trees

Pre-requisites

- Theory of Automaton — DFA, NFA and CFG
- Data Structures — Trees
- Assembly Language — Intermediate code generation

Pre-requisites

- Theory of Automaton — DFA, NFA and CFG
- Data Structures — Trees
- Assembly Language — Intermediate code generation
- Basic Knowledge of several programming languages — useful

If you don’t understand compilers, you can still write programs — you can even be a competent programmer — but you can’t be a master.
— Hal, Abelson, MIT

If you don’t know how compilers work, then you don’t know how computers work. If you’re not 100% sure whether you know how compilers work, then you don’t know how they work.
— Steve Yegge