

# CSCI 447 – Fall 2000

## Compiler Design

Professor : Muhammed F. Mudawwar  
Office : Room 410 FL, x5305  
Office Hours : UTR 10 - 12 or by appointment  
Email : [mudawwar@aucegypt.edu](mailto:mudawwar@aucegypt.edu)  
Web: <http://www.cs.aucegypt.edu/mudawwar/csci447/>  
Textbook: Kenneth Louden, *Compiler Construction: Principles and Practice*, PWS publishing company, 1997  
Reference: Aho, Sethi, and Ullman, *Compilers: Principles, Techniques, and Tools*, Addison Wesley, 1988.

### Objectives

This course presents a practical approach to the subject of compiler construction. It is intended not only to cover the components of a compiler, but also how they actually fit together. The use of compiler tools, such as Lex and Yacc, are emphasized to automate the generation of compiler components, wherever applicable.

### Subjects

- The translation process, major data structures in a compiler, programs related to compilers, TINY sample language and compiler.
- Scanning theory, regular expressions, finite automata, from regular expressions to finite automata, implementing a TINY scanner.
- Using the Lex Scanner Generator.
- Context-free grammars, derivations and parse trees, abstract syntax trees, ambiguous grammars, extended BNF notations, Syntax of TINY.
- Recursive-Descent parsing, LL(1) Parsing, First and Follow sets, Predict function, LL(1) parse table, Recursive-descent parser for TINY.
- Bottom-up parsing, LR parsers, LR(0) items and parsing, SLR(1) parsing, LR(1) and LALR(1) Parsing.
- Using the Yacc parser generator, eliminating ambiguity and conflicts, error recovery, Yacc parser generation for TINY.
- Semantic Processing: attribute grammars, syntax-directed translation, semantic processing techniques.
- Symbol Tables: implementation techniques, symbol attributes, dealing with scope, fields and records.
- Data Types and Type checking.
- Intermediate code, data structures for code generation, basic code generation techniques, using the simple target machine TM.

## Assignments

All assignments can be done in groups of 2 or at most 3 students

- Lex specification for a Pascal-like language
- Yacc specification for a Pascal-like language, handling syntax and semantic errors.
- Type checking and Translation into intermediate code.

## Grading

Written assignments and quizzes: 15%

Programming Assignments: 30%

Midterm Exam: 20% or 25%

Final Exam: 35% or 30%