CS 5900 Compiler Design and Construction

Instructor: Dr. Chen-Fu Chiang

Time: Tuesday 8:40 AM - 11:20 AM
Location: Summit Center 151
Office Hours: Wednesday 3:00 PM - 5:00 PM and Friday 9:00 AM – 12 PM at Summit Center Room **138** or by appointment

Office: WCM 126 B
Email: cchiang@ucmo.edu (preferred and fastest)

Prerequisite: CS 3100 or consent of the instructor

Course Description

I. Purpose of the Course

This course is designed for computer science students and is intended to introduce them to the fundamental concepts of compiler construction. Compiler construction is an important branch of computer science. Studying the parts of a compiler and compiling process is essential in the understanding of any language.

II. Objectives and Desired Student Competencies

Upon completion of this course the student should be able to:

- Understand grammars and finite automata, and languages.
- Implement various phases in the compiling process.

III. Course Content Outline

Text: Compiler Construction Principles and Practices, Kenneth Louden, PWS 1997

A. Introduction
    1. History
    2. Programs Related to Compilers
    3. Major Data Structures in a Compiler

B. Scanning
    1. The Scanning Process
    2. Regular Expressions and Finite Automata
    3. Lex: a Scanner Generator

C. Context-Free Grammar
    1. The Parsing Process
    2. Context-Free Grammars
    3. Parse Trees and Abstract Syntax Trees
    4. Ambiguity
    5. Formal Properties of Context-Free Languages

D. Top-Down Parsing
    1. Top-Down Parsing by Recursive-Descent
    2. LL(1) Parsing
    3. First and Follow Sets

E. Bottom-Up Parsing
    1. Overview of Bottom-Up Parsing
    2. Finite Automata of LR(0) Items and LR(0) Parsing
    3. Yacc: an LALR (1) Parser Generator
    4. Error Recovery

F. Semantic Analysis
    1. Attributes and Attribute Grammars
    2. Algorithms for Attribute Computation
    3. The Symbol Table
    4. Data Types and Type Checking

G. Code Generation
    1. Data Structure for Code Generation
    2. Basic Code Generation Techniques

IV. Procedures/Assessment

The lecture format will be the basic mechanism used in the course. Computer demonstrations in the classroom will be used whenever appropriate. Assessment of student performance will use a criterion referenced model which will include written assignments (30%, might contain programming assignments), regular examinations (midterm 30%) and a comprehensive final exam (40%). Assignments must be turned in at the beginning of the class on the due date. Late assignment will not be accepted due to the short period of summer section. All examinations are closed-book

A typical grading scale will be as follows:

Percent	Grade
90 -100	A
80 - 89	B
70 - 79	C
60 - 69	D
below 60	F

Plagiarism and Cheating of any kind on an examination, quiz, or assignment will ***result at least in an F*** for that assignment (and may, depending on the severity of the case, lead to an F for the entire course). See the UCM Academic Dishonesty Policy at *http://www.ucmo.edu/student/documents/honest.pdf*. I will assume for this course that you will adhere to the academic creed of this University and will maintain the highest standards of academic integrity. In other words, do not cheat by giving answers to others or taking them from anyone else. Make-ups are only given under extreme circumstances. I will also adhere to the highest standards of academic integrity, so please do not ask me to change (or expect me to change) your grade illegitimately or to bend or break rules for one person that will not apply to everyone.