Syllabus for CSE141: Introduction to Compiler Construction

Catalog Description:

This course will examine the basic concepts and techniques in compiler construction. The topics will include lexical analysis, finite automata, parsing, context-sensitive analysis, intermediate representation, code shape analysis, code optimizations, code generation, and register allocation.

Text Books and Other Required Materials:

K.D. Cooper and L. Torczon -- Engineering A Compiler, 2nd Edition.

Course Objectives/ Student Learning Outcomes:

Students will

a. learn the fundamental principles of compilers, the major technologies that support compiler construction, and a basic understanding of the role of compilers in the central activity of computer science.

b. gain hands-on experience in implementing compilers.

c. master skills in solving technical challenges by working on projects.

Program Learning Outcomes:

Students will

a. analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions.

b. design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline.

c. function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline.

d. apply computer science theory and software development fundamentals to produce computing-based solutions.

Course Policies:

Labs are designed to be started and worked on in the time frame you have in lab.

Academic Dishonesty Statement:

a. Each student in this course is expected to abide by the University of California, Merced's Academic Honesty Policy. Any work submitted by a student in this course for academic credit will be the student's own work.

b. You are encouraged to study together and to discuss information and concepts covered in lecture and the sections with other students. You can give "consulting" help to or receive "consulting" help from such students. However, this permissible cooperation should never involve one student having possession of a copy of all or part of work done by someone else in the form of an email, an email attachment file, a diskette, or a hard copy. Should copying occur, both the student who copied work from another student and the student who gave material to be copied will both automatically receive a zero for the assignment. Penalty for violation of this Policy can also be extended to include failure of the course and University disciplinary action.

c. During examinations, you must do your own work. Talking or discussion is not permitted, nor may you compare papers, copy from others, or collaborate in any way. Any collaborative behavior during the examinations will result in failure of the exam and may lead to failure of the course and University disciplinary action.

Course Calendar:

Week 1: introduction; basics

Week 2: lexical analysis

Week 3: lexical analysis; finite automata

Week 4: parser
Week 5: parser
Week 6: parser; context-sensitive analysis
Week 7: context-sensitive analysis; intermediate representation
Week 8: midterm review and exam (tentative)
Week 9: code shape analysis
Week 10: code optimizations
Week 11: code optimizations; code generation
Week 12: code generation

	Week 13: register allocation Week 14: instruction scheduling Week 15: final review and misc.
Class:	T/Th 12:00 – 1:15 PM COB1 288.
Final Exam:	May 10 8:00 – 11:00 AM COB1 288.
Office Hours:	Th 4:00-5:00 PM (Zoom: https://ucmerced.zoom.us/j/87406905790) or by appointment
Assessment/Grading Policy:	Midterm: 20% Final: 35% Lab: 20% Homework: 20% Attendance: 5%