CSE 304
Compiler Design
Course Overview

TONY MIONE

Course Information

- CSE304 – Compiler Design
- Fall 2021
- Instructor: Antonino (Tony) Mione
- Meetings: Tue/Thur : 10:30-11:50 AM

Instructor

- Antonino (Tony) Mione
- Office: B425
- Email: antonino.mione@sunykorea.ac.kr
- Course Website: *http://www3.cs.stonybrook.edu/~amione/CSE304_Course/index.html*
- Phone: +82-032-626-1226
- Office Hours: *Mon/Wed: 11:00-12:00 & 1:00-2:00PM, Tue/Thu: 1:00-2:00 PM*

Announcements

- COVID-19
  - Classes are currently fully online but may switch to in person depending on the COVID situation. [Currently, this does not look likely most or all of this semester.]
  - For classes on campus, masks must be worn at all times in the classroom and seating is assigned by administration to assure social distancing. Please follow these guidelines meticulously.
- Additional video lectures may be noted on the class website. These are recommended for extra instruction to help understand some aspects of compilers.

Misc Information

- Prerequisites:
  - C or higher: CSE 216 or CSE 219 or CSE 260; CSE 220
- Advisory Prerequisites:
  - CSE 303 or CSE 350
- This course gives the student the basics of developing compilers and other language translators. There is a semester long project to give the student the experience of developing a compiler for a small language.

Course Overview

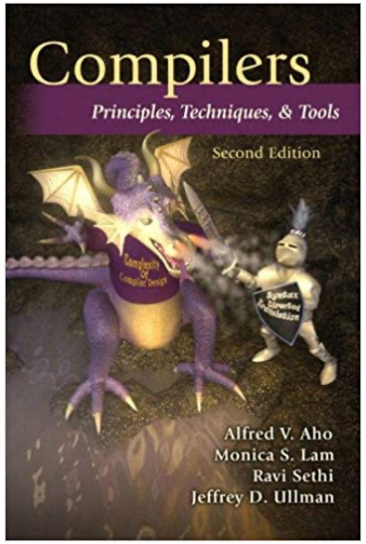
- CSE304
  - Explores the phases of a compiler
  - Examines techniques that are used to analyze high level code and synthesize equivalent low level (machine) code.
  - Explores Run time systems and the management of variables and memory
- Compilers are the tools that allow programmers to create runnable code based on high level ideas and algorithms.
  - Most people can easily learn a new programming language
  - Understanding how a compiler works gives the programmer and software engineer insight into how to develop highly efficient code.

Major Course Topics

- Lexical analysis
- Syntax analysis
- Symbol tables
- Memory allocation
- Code generation
- Semantic analysis
- Abstract interpretation

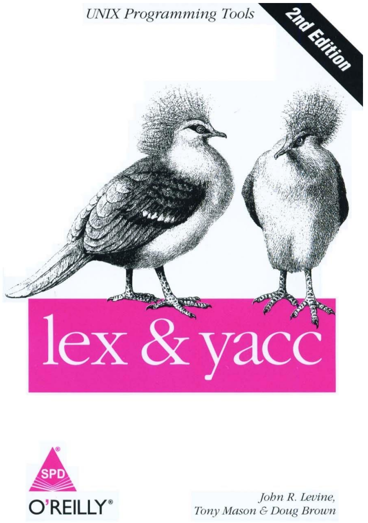
Textbook

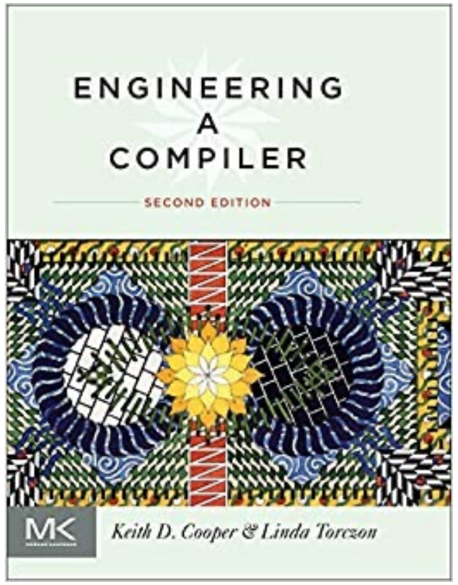
- Required: *"Compilers: Principles, Techniques, and Tools, 2nd Edition"* , Aho, Lam, Sethi, and Ullman, Addison Wesley, 2007. ISBN: 978-0321486813.
- .



Textbook

- Recommended:
  - *“lex & yacc”,* Levine,John, Brown, Doug, and Mason, Tony, O’Reilly, 2012, ISBN 978-1565920002.
  - “Engineering a Compiler”, Cooper, Keith, D., Torczon, Linda, Elsevier, 2012, ISBN 978-0-12-088478-0.





Homework

- Several assignments will be given either from book problems or problems I create.
- These assignments will be to help you practice and understand theoretical concepts presented in class.
- Problems are due on fixed dates and times
- All work will be completed on an ***individual basis*** (write your own code) *unless otherwise instructed!*
- You will use Blackboard to submit your completed assignments
- The usual drill: Start homework assignments as soon as they are given. I try to give more than enough time to complete these but they hold some challenges and working last minute will add considerable stress.

Late Homework Policy

- Assignments must be turned in by the due date and time.
- Any part of an assignment that's late means the entire assignment is late.
- If your assignment is incomplete or not entirely working by the due date, turn in what you have to get some partial credit.
- If you have an emergency situation, email me before the due date and I may be able to work something out (case-by-case basis).
- A penalty may be applied to the score.

Bottom line: Plan ahead, start early!

Term Project

- The Project for the class will involve writing a compiler for a small subset of C
- It will be rolled out in phases matching the different phases of a compiler:
  - Lexical Analyzer
  - Symbol Table Manager
  - Parser
  - Semantic Actions/Intermediate Code Generation
  - Code Generator
- We will likely only have time for the 1st 4 phases (we'll see)
- Coding may be done in *any language that you are comfortable with* ***as long as I am able to build the code with my equipment/os/language processor.***
  - Suggested Environment: Windows 10 or Ubuntu
  - Suggested Languages: C, C++, Java, Python, etc

Cooperation vs. Copying

- Cooperation (talking over problems) is a good way to learn and is encouraged
- ***Do not copy code. Do not let others look at or copy your code.***
- Copying is not allowed on homework or exams no matter the source (written or verbal)
- When you submit your homework or tests:
  - **You are pledging that the work is your own and you have not copied it.**
  - **You are also pledging that you have not allowed others to copy it.**

Examinations

- Examination dates will be posted on the schedule page of the course website. Tentative dates are:
  - Midterm exam 1: Wed, 10/7
  - Midterm exam 2: Wed, 11/11
  - Final exam: See SUNY Korea Final Exam schedule
- Do not miss exams
  - Arrange your work and travel schedules as needed to be present for examinations
  - Makeup exams will only be given for verified, officially sanctioned university activities. Makeup examinations may be oral.

Examinations

- For Examinations given in class:
  - examinations will be closed-notes and closed-book, except one sheet of notes (A4 or 8.5x11), both sides, handwritten
- For Examinations given on Zoom:
  - You are allowed the use of your book and class slides (even pdfs)
  - You must be on Zoom in the class lecture room
  - You must have your web cam on pointed at yourself during the test

Grading

Problem Sets: 25%

Project:30%

Midterm Exam 1: 10%

Midterm Exam 2: 10%

Final Exam: 20%

Class Participation (Attendance): 5%

Re-Grading

- If you feel that your work (exam, homework, etc.) was not graded correctly, you may request a regrade no later than one week from the day grades are posted or announced
- Requests for regrades made after one week will not be entertained

Electronics in Class

- Cell phones should be put away during class
- Laptops may be used during periods where you are asked to work on an exercise during class
- Lecture slides are available on the course website for study before class.
- Talk to me after class if there's an issue with this policy

Disability

- If you have a physical, psychological, medical or learning disability, please contact the One-Stop Service Center.
  - Location: Academic Building A201
  - Phone: 626-1117
- The DSS will determine with you what accommodations, if any, are necessary and appropriate
- All information and documentation of disability is confidential

How to Succeed in this Class

- Attend class and be on time!
- Not all information is in my lecture notes or in the book
- I sometimes do in-class demos that emphasize non-obvious details
- The project work will take a lot of your time, so practice good time management
- Read the reading assignments and review the lecture notes
- Those who create mini-projects and write extra code to explore concepts that were not assigned ("for fun") generally do best.
- Ask questions right away if confused. Ask in class, come to my office hours or send email. Don't stay confused and don't get behind! (This class can get ahead of you quickly!)
- Welcome and I hope you enjoy the class and learn a lot!

Questions