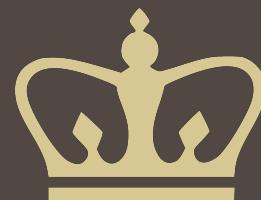


PROGRAMMING LANGUAGES & TRANSLATORS

Instructor: Baishakhi Ray

Fall 2023

COMS 4115: Trivia



Instructor

Prof. Baishakhi Ray

Associate Professor

rayb@cs.columbia.edu

<https://rayb.info>

Office Hours: Monday noon to 1 pm/by Appointment

Location: CEPSR 6LE1



PLT 4115

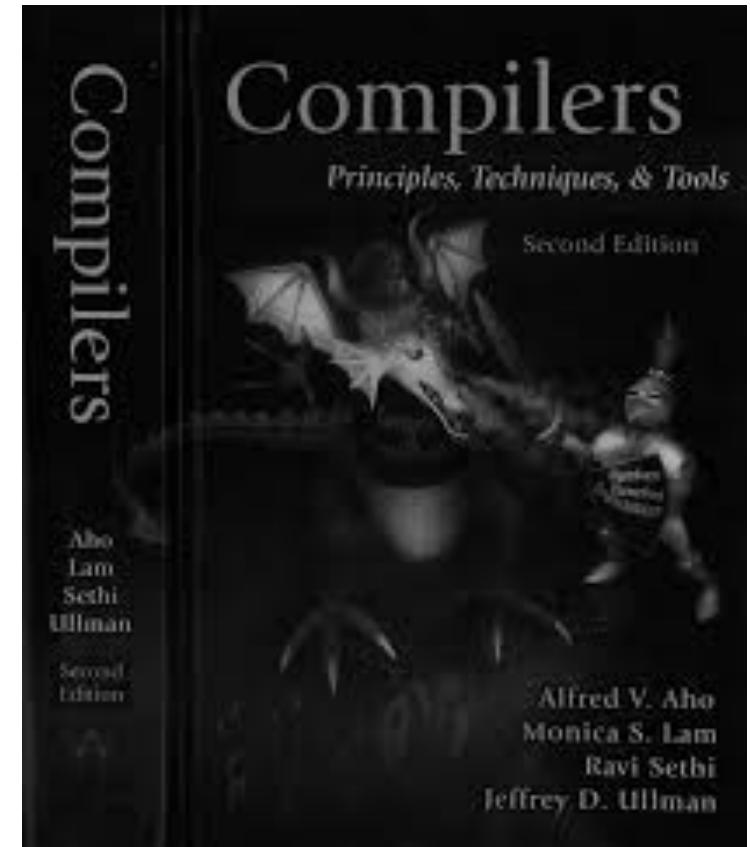
- Lectures:
 - Mondays and Wednesdays, 10:10 AM-11:25 AM @ CSB 451
 - Jan 18 – May 1
- All the class lectures will be recorded by CVN and will be available in video library tabs in CVN.
- Get all the class updates on the website
 - <https://www.rayb.info/plt4115-spring2023>
- We will use Ed Discussion for class communication
 - See your coursework tab option

Programming Language & Translators

How can a computer program written in a high-level **programming language** (e.g., C, Python) be **translated** to a lower-level language (e.g., assembly language or machine code) to create an executable program?

Recommended Text

- Compilers: Principles, Techniques, and Tools
 - By Alfred V. Aho, Monica S. Lam, Ravi Sethi, and Jeffrey D. Ullman.
 - 2nd Edition
 - Addison-Wesley, 2006
- We will follow this book, but not line-by-line/section-by-section



This Class

- Theory: Learn different phases of a compiler design (50%)
- Practice: Implement different phases of compilers (50%)
 - Implement modules on top of an existing compiler (Clang/LLVM)

	Lectures	Programming
1	Introduction	
2	Lexical Analysis	Prog1
3	Syntax Analysis	Prog2
4	Semantic Analysis	Prog3
5	Run-Time Environment	
6	Code Generation	Prog4
7	Optimization	Prog5, Prog6

Theory deliverables:

- Written assignments
- Midterm
- Final

Programming deliverables:
6 prog assignments

All are individual assignments

Assignments and Grading

• Programming Assignments	50%
• Written Assignments	10%
• Midterm	20%
• Final	20%
• Extra Credit	10%

- Programming assignments are most important, but most students do well on it. Grades for tests often vary more.

Extra Credit:

- 10% of earned (extra credit/total extra credit) will be added with the original 100% from other assignments/exams
 - If you earn 50 out of 100 in extra credit, 5 will be added with your total (100%) achievement.

Assignments Policy

- Hard Deadline

- There will be no extension unless you produce medical certificate or permission from school authorities
- The instructor or TAs will not reply to such email requests.
- Plan ahead so that you can finish the assignments on time.
 - There can be challenges that you have not anticipated

- Written Assignments will be submitted through Gradescope

- We will share Gradescope entry code
- Type your submission

- Programming Assignments will be submitted through Github Classroom

- TAs will send you detailed instructions

Assignments Policy

- Work individually
- Programming assignments: work individually.
 - You can discuss with TAs/Instructor/Classmate
- Written assignments: do by yourself.
 - No discussion
 - Only clarification questions are allowed on Ed Discussion
 - TAs/Instructors will not respond to individual email
- DO NOT USE AI-Assisted Tool.
 - You will not learn
 - We will check for plagiarism

Submission Policy

Don't be a cheater (e.g., copy from each other).
If I catch you cheating I will send you to the dean.

- Read the CS Department's Academic Honesty Policy: <https://www.cs.columbia.edu/education/honesty/>
- **OK:** Discussing lecture content
- **Not OK:** Solving a homework problem with classmates
- **OK:** Doing programming assignments together
- **Not OK:** Copying from others' solutions.
- **Not OK:** Posting any homework questions or solutions.
- **Not OK:** Use AI-assisted tools to find the answers.

Exam Policy

- Exams: Open book
 - Follow CU honor code.

- In-Class Participations
 - Class participation is important
 - Instructor will ask questions that you have to answer

Prerequisites

1. Advanced Programming on C/C++
2. Computer Science Theory
 1. Regular languages and expressions
 2. Context-free grammars
 3. Finite automata (NFAs and DFAs)
3. Fundamentals Of Computer Systems
 1. Memory layout
 2. Register
 3. Instruction Set
 4. Performance Analysis

We will **conduct a take-home exam** on the first day of the class (18th January).

Entrance TakeHome Exam

- Due on 22nd **Jan** 11:59 pm (Hard Deadline)
 - 75 minutes (timed exam)
 - We will send you the exam link. Once you start working on it, it has to be finished in 75 minutes.
- Open Book/Open Internet---no AI Assisted tool.
- Only pass/fail to check your ability to test the class
 - Grades of the entrance exam **will not be** added to the final course grade
- We will email a Google Form link right before the exam.
- Use your **uni email** to sign in to the Google form
 - This will help us to track you
- All the students have to pass the exam to take the class.

Entrance Exam

- Don't be a cheater
 - It's an open book/open internet
 - You are not allowed to discuss with anyone
- Read the CS Department's Academic Honesty Policy: <https://www.cs.columbia.edu/education/honesty/>
- If you cheat to pass the exam, it will ultimately be your loss, as you will have immense difficulty taking the course
 - Instructors or TAs will not entertain questions that you already are supposed to know

Exam Schedule

- Midterm: March 6
- Final: May 1 (Last Day of Class)

Submission Links

- **Written Assignments** : [gradescope](#)
Entry Code will be posted in Coursework
- **Programming Assignments** : [github classroom](#)
Details will be posted in Coursework

Q&A

Programming Assignments/Projects

- We will implement some compiler functionalities through out the semester
- Three projects -- you can pick any one of the following three options
 - Dead-code elimination
 - Finding resource leak
 - Finding uninitialized variables
- We will implement the compiler functionalities step-by-step to achieve the final goal.
 - There will be 6 programming assignments that will help you to reach the goal

Programming Assignments

- 6 programming assignments
- We should have a good knowledge of C/C++
- Linux is preferred operating system
 - All the instructions will be given based on Linux

Programming Assignments

- The assignments will be done on top of LLVM infrastructure
 - LLVM is a state-of-the art compiler (default compiler in MAC)
 - Some of the assignments are inter-dependent
 - We will provide Google cloud resource for LLVM related assignments
- Each student will be provided with google cloud resource for programming assignment.

Programming Assignments

- In all the assignments, some partial code will be given.
- Based on the instructions, you have to finish the rest.
- Scripts to setup the environment and run the code will also be provided.
- Each assignment will come with a bunch of test cases
 - Your goal is to pass all the test cases.
 - However, passing all the test cases do not mean the program is correct.
- Submit all programming assignments through GitHub Classroom.

Written Assignments

- After every topic, there will be a written assignment
 - Q&A
 - Multiple Choice
 - Problem Solving
- Midterm and Finals will follow written assignments patterns

TA

- Vikram Nitin.
- Ziyuan Zhong.
- Alexis Gadonneix

Q&A