Computer Systems
CSCI 2400 Syllabus

Course Overview

Course covers how programs are represented and executed by modern computers, including low-level machine representations of programs and data, an understanding of how computer components and the memory hierarchy influence performance.

General Information

	Section: 100,110	Section: 200,210
Instructor(s):	Maciej Zagrodzki	David Knox
Contact(s):	`maciej.zagrodzki@colorado.edu`	`david.knox@colorado.edu`
Office Hours: (via Zoom)	Tu 9-11am MT	Tu, Th: 6 - 7pm MT
Zoom link:	**https://cuboulder.zoom.us/j/** 93938506998	**https://cuboulder.zoom.us/j/** 490423339

Prerequisites & Materials

Prerequisites: CSCI 2270 or CSCI 2275

Primary Text: *Computer Systems, A Programmer's Perspective, 3rd Edition (Bryant and O'Halloran).*
Errata: `https://csapp.cs.cmu.edu/3e/errata.html`
The international edition is acceptable and much cheaper ($30) (Amazon link) - Just be aware that there are mistakes in some practice problem solutions in the international book.
ISBN-13: 978-0134092669
ISBN-10: 013409266X

Supplementary Text: There will be several handouts posted in Canvas.

Canvas link: **https://canvas.colorado.edu**
The course uses Canvas website for links to course content, such as video links, lecture notes, assignments, quizzes, grading interview schedulers, and grades.

Requirements for COVID-19

There is a mandatory mask requirement for attending in-person lectures this semester. All students will need to conform to strict physical distancing when in the lecture room. Everyone has the responsibility as CU Boulder students to complete the Daily Health Form any day they are coming to campus.

As a matter of public health and safety due to the pandemic, all members of the CU Boulder community and all visitors to campus must follow university, department and building requirements, and public health orders in place to reduce the risk of spreading infectious disease. Required safety measures at CU Boulder relevant to the classroom setting include:

- maintain 6-foot distancing when possible,
- wear a cloth face covering (over nose and mouth), especially when unable to maintain a distance of at least 12 feet,
- clean local work area,
- practice hand hygiene,
- follow public health orders,
- if sick and you live off campus, do not come onto campus (unless instructed by a CU Healthcare professional),
- if sick and you live on-campus, please alert CU Boulder Medical Services.

Students who fail to adhere to these requirements will be asked to leave class, and students who do not leave class when asked or who refuse to comply with these requirements will be referred to Student Conduct and Conflict Resolution. For more information, see the policies on COVID-19 Health and Safety and classroom behavior and the Student Code of Conduct. If you require accommodation because a disability prevents you from fulfilling these safety measures, please see the "Accommodation for Disabilities" statement on this syllabus.

Before returning to campus, all students must complete the COVID-19 Student Health and Expectations Course. Before coming on to campus each day, all students are required to complete a Daily Health Form. In this class, you may be reminded of the responsibility to complete the Daily Health Form and given time during class to complete it.

Students who have tested positive for COVID-19, have symptoms of COVID-19, or have had close contact with someone who has tested positive for or had symptoms of COVID- 19 must stay home and complete the Health Questionnaire and Illness Reporting Form remotely. In this class, if you are sick or quarantined, notify us immediately. The instructor and TA will meet with you to determine the best plan to complete all work for the course depending on your ability to perform class work during your quarantine or recovery.

Email Policy

We would prefer that all communications about the course be through the Piazza Forums. If you need to send a private message to the instructor, Piazza allows you to make your post private to a specific instructor. If you must use e-mail, note that we probably won't respond outside of business hours. Also, put CSCI 2400 somewhere in the subject and try to keep the e-mail brief. If it is ends up being more than two paragraphs, save it for office hours.

Canvas

We will use Canvas as our course website. Course content, such as lecture notes, assignments, quizzes, interview schedulers, and exams will be posted here. The course page can be found by navigating to `https://canvas.colorado.edu` then *CSCI 2400* under the *Course* tab.

Github

Github will be used for distributing and submitting the lab assignments. For each assignment, a unique URL will be posted. Every student will be required to follow the URL and accept the assignment. Accepting the assignment will automatically generate a private Github repository with starter included code. Students are encouraged to push their changes to their repository **frequently** as they make progress on the assignment.

Piazza

- All official course announcements and communication will be done through Piazza.
- Instructor(s) will *not* immediately respond to Piazza questions about being stuck on a programming assignment. These sorts of questions should be asked during office hours. Students are welcome to nudge their colleagues in the right direction (which we will later endorse if it is good advice), but nobody should be giving away answers.
- Instructor(s) will respond ASAP to Piazza posts if they are emergencies (e.g., Canvas not working, exam not working, etc..).
- Instructor(s) don't answer Piazza over the weekend (6PM Friday to 9AM Monday).

What is Piazza?

It is like a classroom, it is a place to ask questions, discuss learning strategies, explore related topics, support your classmates, and contribute to the class. It is also the primary means for an instructor to communicate with the whole class in course information and announcements.

What Piazza is not?

It is NOT a 24-hour helpdesk or answer forum. It is NOT a robot tutor. It is NOT a personal blog space. It is NOT a place for others to completely debug your code.

Zoom

Office hours will be conducted via zoom. If instructor(s) are 10 minutes late for office hours, send an e-mail to the instructor(s). If no instructor present 20 minutes into office hours, then office hours are officially canceled and a make up time will be arranged.

What are ZOOM Office Hours?

It is much like a classroom. It is a live web session to ask questions, explore further, discussion strategies, explore related topics and support your classmates and contribute to the class. Office hours are optional.

How do we use ZOOM Office Hours?

Discuss class content directly with the instructor and other students. Practice professional collaboration strategies. Instructors may use break out groups or/and guide discussions, at their discretion. Instructors decide which topics will be discussed based on what will optimize learning. Please wear clothes and be aware of your environment.

What are ZOOM Office Hours not?

It is NOT an Answer Forum. It is NOT a Tutoring Session. Instructors cannot help you completely debug your code. The instructor may not reply to all inquiries and let other students speak as appropriate.

Piazza and Zoom Online Conduct

Posts and questions about classwork should be content specific and reflect an effort by the student's asking the question. Here are sample questions that are not content-specific and hence not appropriate:

- Tell me how to do #6
- Please explain #6
- I'm lost on #6, Help!

Sample questions that are content-specific:

- I applied the technique from the video to #6, but I get an answer that is too large, could my loop be incorrect?
- In the video lecture, I understand the algebra steps in #6, but why is 0! =1? Is anyone else getting different answers to #6? It seems to depend on which method I use. (notice the student is not sharing the solution)

Any behavior that interferes with student learning or university activities is not acceptable.

Any behavior, regardless of intent, that could cause abuse, interference, obstruction, disruption, or interference with student learning, including comments that are disrespectful, aggressive, distracting, off-topic or inappropriate, will not be tolerated and may result in being muted, at the instructor's discretion. Such behavior will be reported to Student Conduct and Conflict Resolution. For example, dominating group discussions can inhibit others opportunity to learn and will not be allowed.

Course Conduct and Collaboration

- **We do *not* want anyone to use Google, GitHub, or StackOverflow to get ideas for coding the programming assignments.** We understand that you may have taken courses where you are allowed to do this, under the condition that you provide the proper citations, but this is forbidden in this course. This course is designed to be *self-contained*, so there is no need to use outside material from the internet to complete the assignment (unless of course you are explicitly told to research a particular part of the programming assignment online).
- **However, you may use Google or StackOverflow for things that do not involve the *creation* of code** such as deciphering cryptic `gcc` warning and error messages, or reminding ourselves about certain aspects of `C` and Linux.
- You are allowed to discuss the programming assignments with your colleagues on Piazza; however, **there should be no exchange of code or detailed pseudocode whatsoever**. In other words, natural languages should be used to communicate about the assignments, not formal languages.

Any violation of the rules listed above constitutes cheating. If cheating occurs, you will receive a zero for the assignment and your case will be sent to the Honor Code Council.

Learning goals

- Explain and perform common logical operations (and, or, negation, conversion) on binary variables and binary vectors and identify and apply common boolean algebraic laws such as DeMorgan's laws, idempotence, etc
- An ability to translate between integer binary and decimal data, detect and identify the outcome of operations due to limited data representations (e.g. overflow), distinguish between the data representations and ranges for signed and unsigned data types Translate IEEE floating point representation to and from binary and real numbers and identify the limitations of fixed-precision floating point representation.
- An ability to related compiler-generated assembly programs to the corresponding higher level language structures with sufficient ability to enable debugging high-level programs. Given a machine language representation of a program compiled in a higher level language, students should be identified and describe the operation of conditional statements, loops, function calls, switch statements.
- The ability to explain how higher level language functions are implemented using the stack of an underlying machine, including how local variables are allocated, trace the execution due to recursion and identify and trace the effect of buffer-overflow of the stack.
- An ability to explain how high-level program structures can be restructured to facilitate optimization for pipelined architectures and cache memory hierarchies.
- An ability to explain how computer memory is organized and represented both to the programmer and to the computer architecture by the operating system through the use of virtual memory mapping.
- An understanding of how to use asynchronous signals, concurrent programs, and the programming issues that arise with such programs, such as race conditions. Identify and construct processes on a common computer platform, identify and perform basic synchronization between processes and understand the costs and benefits of using processes.
- An ability to explain how global memory, function-local and dynamic memory allocation is performed and the performance benefits of each form of memory allocation.
- An ability to explain how programming errors may affect program correctness, including errors in function calls, memory allocation, integer, and floating point data representations.
- An ability to measure program performance and use that measured information to determine how to improve program performance.
- An ability to use a machine-level debugger and inspect the memory and register state of programs.

Grading Components

This is a **4 credit** course, so please be prepared to put in a significant amount of effort, and make sure that you are getting started early on all assignments.

The course has 3 grading components with the following breakdown:

- Study and reflection questions (weekly quizzes) - 8% of grade
- Four non-cumulative exams - 8% each, 32% of grade
- Six programming labs - 10% each, a total of 60% of grade

Reading Quizzes: You will be assigned reading questions similar to the practice questions from the text. Most reading quizzes will be automatically graded and are designed to help you rectify your knowledge of the material in a particular section of the book. You should also attempt the self-study questions in the book (answers at the end of the section).

Exams: You will have four exams. Each of the exams covers significant material usually over multiple chapters of the book.

Lab Assignments: Your primary assignments will be your "Lab Assignments," given every 2-3 weeks, each of which will be followed by a grading meeting to review your solution with the instructor or TA. The labs are the primary learning vehicle for this course. The grade meetings are scheduled on the Canvas site typically just before the assignment is due, and will begin after the due-date of each Lab. Even if you complete your work, you will not receive a grade unless you conduct a grading session.

We have six Labs:

Data Lab	The purpose of this assignment is to become more familiar with bit-level representations of integers (and floating point) numbers. You'll do this by solving a series of programming "puzzles." Many of these puzzles are quite artificial, but you'll find yourself thinking much more about bits in working your way through them.
Bomb Lab	The nefarious Dr. Evil has planted a slew of "binary bombs" on our class machines. A binary bomb is a program that consists of a sequence of phases. Each phase expects you to type a particular string on stdin. If you type the correct string, then the phase is defused and the bomb proceeds to the next phase. Otherwise, the bomb explodes by printing "BOOM!!!" and then terminating. The bomb is defused when every phase has been defused. There are too many bombs for us to deal with, so we are giving each student a bomb to defuse. Your mission, which you have no choice but to accept, is to defuse your bomb before the due date. Good luck, and welcome to the bomb squad!

Attack Lab	The Attack Lab involves generating a total of five attacks on two programs having different security vulnerabilities. You will learn different ways that attackers can exploit security vulnerabilities when programs do not safeguard themselves well enough against buffer overflows.
Performance Lab	This assignment deals with optimizing memory intensive code. Image processing offers many examples of functions that can benefit from optimization. In this lab, you will be improving the overall performance of an "image processing" application by a factor of about 25 – an extra credit phase will challenge you do increase the speed further. In the past, some students have sped up the code by a factor of 200-300.
Shell Lab	The purpose of this assignment is to become more familiar with the concepts of process control and signalling. You will do this by writing a simple Unix shell program that supports job control. You may work in a group of up to two people in solving the problems for this assignment.
Malloc Lab	In this lab you will be writing a dynamic storage allocation for C programs (i.e., your own version of the *malloc, free* and *realloc* routines). You are encouraged to explore the design space creatively and implement an allocator that is correct, efficient and fast. You may work in a group of up to two people.

The grades for each lab will be based 40% upon the Task Success (i.e. "does it work") and 60% upon your explanation of your code/assignment and answering questions about the lab and its concepts. Historically speaking, students that have completed the assignment themselves usually have a little problem passing the Q & A portion of the grade.

Students may work in teams of up to two for the labs only, but each student will still be responsible for scheduling their own grading meeting with the TA for each lab. You may help others only to the extent of answering typical questions that arise during compiling, debugging, and executing your lab assignments.

All assignments are due by the deadline stated. Extensions will not be granted except at the instructor's discretion in documented cases of extreme hardship, emergency, etc., unless otherwise noted.

On the task success 40% portion of the grade, we strongly encourage you to submit even partially working code/assignment by the deadline to obtain partial credit on the 40% for task success. You must attend your grading meeting to qualify for grading points. If you miss your meeting with the TA (without notifying your TA ahead of time with a suitable reason), this may result in a zero grade for the assignment. The TA is under no obligation to reschedule your appointment if you miss your meeting, so write down your meeting times, and don't forget them! Even if you are unable to submit fully working code/assignment by the deadline, we strongly encourage you to keep working at a full solution for the assignment, which should benefit your understanding and ability to answer questions during the Q & A

meeting with the TA. All labs must be written in C and compiled for execution on the class programming environment, unless otherwise noted.

Exam Dates Fall 2020

Exam #1	Chap 1,2	Sept 18
Exam #2	Chap 3	Oct 16
Exam #3	Chap 4,5,6	Nov 6
Exam #4	Chap 7,8,9	Section 100: Wednesday, Dec. 9, 1:30–4 p.m.
		Section 200: Thursday, Dec. 10, 1:30–4 p.m.

Lab Due Dates Fall 2020

Data Lab	Monday 11:00pm	Sept 14th
Bomb Lab	Monday 11:00pm	Sept 28th
Attack Lab	Monday 11:00pm	Oct 12th
Performance Lab	Monday 11:00pm	Nov 2nd
Shell Lab	Monday 11:00pm	Nov 16th
Malloc Lab	Monday 11:00pm	Nov 30th

Important Fall 2020 Term Dates

Classes start:	August 24th
Labor Day (University Closed):	September 7th
Thanksgiving Holiday (University Closed):	November 26 - 27th
Last Day of Classes:	December 7th
Final Exam Period:	December 9 - 13th

The following is a tentative schedule for our course.

- **Week 1: Aug 24 - 28**
  - *Chapter 2.1 - Information Storage*
  - *Chapter 2.2 - Information Representations*
  - *Recitation*: Start of the Data Lab
- **Week 2: Aug 31 - Sept 4**
  - *Chapter 2.3 - Information Arithmetic*
  - *Chapter 2.4 - Floating Point*
  - *Recitation*: Data Lab trouble shooting, start GDB tutorial
- **Week 3: Sept 8 - 11**
  - *Chapter 3.1 - Historical Prespective*
  - *Chapter 3.2 - Program Encodings*
  - *Chapter 3.3 - Data Formats*
  - *Chapter 3.4 - Accessing Information*
  - *Recitation*: start Bomb Lab, more GDB
- **Week 4: Sept 14 - 18**
  - ***Data Lab DUE*: Monday Sept 14, 11 pm** *(Interviews Tue-Fri)*
  -
  - *Chapter 3.5 - Arithmetic and Logical*
  - *Chapter 3.6 - Control*:
  - *Chapter 3.7 - Procedures*:
  - *Recitation*: Bomb Lab Trouble Shooting
  - ***Exam 1*: Friday Sept 18th**

- **Week 5: Sept 21 - 25**
  - *Chapter 3.8 - Array Allocation and Access*
  - *Chapter 3.9 - Heterogeneous Data Structures*
  - *Chapter 3.10 - Combining Control and Data*
  - *Chapter 3.11 - Floating Point Code*
  - *Recitation*: Bomb Lab completion
- **Week 6: Sept 28 - Oct 2**
  - ***Bomb Lab Due*: Monday Sept 28, 11 pm** *(Interviews Tue-Fri)*
  - *Chapter 5.1 - Capabilities and Limitations of Compilers*
  - *Chapter 5.2 - Expressing Program Performance*
  - *Chapter 5.3 - Program Example*
  - *Chapter 5.4 - Eliminating Loop Inefficiencies*
  - *Chapter 5.5 - Reducing Procedure Calls*
  - *Chapter 5.6 - Eliminating Unnecessary Memory References*
  - *Chapter 4.3 - Sequential Processors*
  - *Chapter 4.4 - Pipelined Processors*
  - *Recitation*: Attack Lab
- **Week 7: Oct 5 - 9**
  - *Chapter 5.7 - Understanding Modern Processors*
  - *Chapter 5.8 - Loop Unrolling*
  - *Chapter 5.9 - Enhanced Parallelism*
  - *Chapter 5.10 - Summary of Optimizations*
  - *Chapter 5.11 - Some Limiting Factors*
  - *Chapter 5.12 - Understanding Memory Performance*
  - *Chapter 5.13 - Life in the Real World*
  - *Chapter 5.14 - Identifying and Eliminating Performance Bottlenecks*
  - *Recitation*: Attack Lab trouble shooting

- **Week 8: Oct 12 - 16**
  - ***Attack Lab Due*: Monday Oct 12, 11 pm** *(Interviews Tue-Fri)*
  - *Chapter 6.1 - Storage Technology*
  - *Chapter 6.2 - Locality*
  - *Chapter 6.3 - The Memory Hierarchy*
  - *Chapter 6.4 - Cache Memories*
  - *Chapter 6.5 - Writing Cache Friendly Code*
  - *Chapter 6.6 - Putting It All Together*
  - *Recitation*: Start Performance Lab
  - ***EXAM #2*: Friday Oct 16th**
- **Week 9: Oct 19 - 23**
  - *Chapter 8.1 - Exceptions*
  - *Chapter 8.2 - Processes*
  - *Chapter 8.3 - System Call Error Handling*
  - *Chapter 8.4 - Process Control*
  - *Recitation*: Performance Lab trouble shooting
- **Week 10: Oct 26 - 30**
  - *Chapter 8.5 - Signals*
  - *Chapter 8.6 - Nonlocal Jumps*
  - *Chapter 8.7 - Tools for Manipulating Processes*
  - *Recitation*: Shell Lab Orientation
- **Week 11: Nov 2 - 6**
  - ***Performance Lab Due*: Monday Nov 2, 11 pm** *(Interviews Tue-Fri)*
  - *Chapter 9.1 - Virtual Memory - Physical and Virtual Addressing*
  - *Chapter 9.2 - Address Spaces*
  - *Chapter 9.3 - Caching*
  - *Chapter 9.4 - Memory Management*
  - *Chapter 9.5 - Memory Protection*
  - *Recitation*: Shell Lab Issues
  - ***EXAM #3*: Friday Nov 6th**

- **Week 12: Nov 9 - 13**
  - *Chapter 9.6 - Address Translation*
  - *Chapter 9.7 - Case Study*
  - *Chapter 9.8 - Memory Mapping*
  - *Chapter 9.9 - Dynamic Memory Allocation*
  - *Chapter 9.10 - Garbage Collection*
  - *Chapter 9.11 - Common Memory-Related Bugs*
  - *Recitation*: Malloc Lab Orientation
- **Week 13: Nov 16-20**
  - ***Shell Lab Due*: Monday Nov 16, 11 pm** *(Interviews Tue-Fri)*
  - *Chapter 7.1 - Compiler Process*
  - *Chapter 7.2 - Static Linking*
  - *Chapter 7.3 - Object Files*
  - *Chapter 7.4 - Relocatable Object Files*
  - *Chapter 7.5 - Symbols and Symbol Tables*
  - *Recitation*: Malloc Lab Issues
- **Week 14: Nov 23-25 (Thanksgiving Break)**
  - *Chapter 7.6 - Linking with Libraries*
  - *Chapter 7.7 - Relocation*
  - *Chapter 7.8 - Executable Object Files*
  - *Chapter 7.9 - Loading Executable Object Files*
  - *Recitation*: Malloc Lab issues
  - ***short week, two recitations(206, 207) and one lecture (on Thursday or Friday) will be lost***

- **Week 15: Nov 30 - Dec 4**
  - ***Malloc Lab Due*: Monday Nov 30, 11 pm** *(Interviews Tue-Fri)*
  - *Chapter 7.10 - Dynamic Linking with Shared Libraries*
  - *Chapter 7.11 - Loading and Linking Shared Libraries*
  - *Chapter 7.12 - Position-Independent Code (PIC)*
  - *Chapter 7.13 - Library Interpositioning*
  - *Recitation*: Student led Exam Review
- **Week 16 - Last day of classes Dec 7 - Final Exam Period: Dec 9 - 13**
  - *Recitation*: Student led Exam Review
  - *Final Schedule still being formulated: Last Fall it was on the 2nd and 3rd day*
  - ***Section 100 : EXAM #4*: Wednesday, Dec. 9, 1:30–4 p.m.**
  - ***Section 200 : EXAM #4*: Thursday, Dec. 10, 1:30–4 p.m.**

Classroom Behavior

Students and faculty each have responsibility for maintaining an appropriate learning environment. Those who fail to adhere to such behavioral standards may be subject to discipline. Professional courtesy and sensitivity are especially important with respect to individuals and topics dealing with race, color, national origin, sex, pregnancy, age, disability, creed, religion, sexual orientation, gender identity, gender expression, veteran status, political affiliation or political philosophy. Class rosters are provided to the instructor with the student's legal name. we will gladly honor your request to address you by an alternate name or gender pronoun. Please advise us of this preference early in the semester so that we may make appropriate changes to my records. For more information, see the policies on classroom behavior and the Student Code of Conduct.

Accommodation for Disabilities

If you qualify for accommodations because of a disability, please submit your accommodation letter from Disability Services to your faculty member in a timely manner so that your needs can be addressed. Disability Services determines accommodations based on documented disabilities in the academic environment. Information on requesting accommodations is located on the Disability Services website. Contact Disability Services at 303-492-8671 or dsinfo@colorado.edu for further assistance. If you have a temporary medical condition, see Temporary Medical Conditions on the Disability Services website.

Preferred Student Names and Pronouns

CU Boulder recognizes that students' legal information doesn't always align with how they identify. Students may update their preferred names and pronouns via the student portal; those preferred names and pronouns are listed on instructors' class rosters. In the absence of such updates, the name that appears on the class roster is the student's legal name.

Honor Code

All students enrolled in a University of Colorado Boulder course are responsible for knowing and adhering to the Honor Code. Violations of the policy may include: plagiarism, cheating, fabrication, lying, bribery, threat, unauthorized access to academic materials, clicker fraud, submitting the same or similar work in more than one course without permission from all course instructors involved, and aiding academic dishonesty. All incidents of academic misconduct will be reported to the Honor Code (honor@colorado.edu); 303-492-5550). Students found responsible for violating the academic integrity policy will be subject to nonacademic sanctions from the Honor Code as well as academic sanctions from the faculty member. Additional information regarding the Honor Code academic integrity policy can be found at the Honor Code Office website.

Sexual Misconduct, Discrimination, Harassment and/or Related Retaliation

The University of Colorado Boulder (CU Boulder) is committed to fostering an inclusive and welcoming learning, working, and living environment. CU Boulder will not tolerate acts of sexual misconduct (harassment, exploitation, and assault), intimate partner violence (dating or domestic violence), stalking, or protected-class discrimination or harassment by members of our community. Individuals who believe they have been subject to misconduct or retaliatory actions for reporting a concern should contact the Office of Institutional Equity and Compliance (OIEC) at 303-492-2127 or cureport@colorado.edu. Information about the OIEC, university policies, anonymous reporting, and the campus resources can be found on the OIEC website.

Please know that faculty and instructors have a responsibility to inform OIEC when made aware of incidents of sexual misconduct, dating and domestic violence, stalking, discrimination, harassment and/or related retaliation, to ensure that individuals impacted receive information about options for reporting and support resources.

Religious Holidays

Campus policy regarding religious observances requires that faculty make every effort to deal reasonably and fairly with all students who, because of religious obligations, have conflicts with scheduled exams, assignments or required attendance. In this class, Faculty: insert your procedures here.

See the campus policy regarding religious observances for full details.