Welcome!

You’ve been writing programs for the past year (or more) and you have become more capable of analyzing the efficiency of programs (CSE 12 and 21 or equivalents) and designing more advanced programs (CSE 12 or equivalent). However, the computer likely remains nearly a black box for you. How does the code you write get converted into instructions the machine can interpret? What do those instructions look like and how are they managed? Moreover, how does the computer execute those instructions?

This is the first course in a series of courses in our program that aims to demystify computing so you become more capable and knowledgeable computer scientists. In CSE 30, you’ll learn about how low-level programming works to prepare you for later courses in our curriculum that heavily leverage this knowledge, including CSE 100, CSE 120, CSE 131, CSE 140, CSE 141, and CSE 142. Ultimately, the ideas you’ll learn in this class are critical for your success as a computer scientist and we hope you look forward to diving deeper into systems this quarter.

What does this course offer to you?

This course is designed to convey three critical ideas in computing. First, you’ve spent most of your time programming in Java and while Java (or Python) makes programming easier for you, it hides away important details of how programs really work that impact their performance (namely how pointers function and how memory is managed). We’ll learn these ideas in depth in the programming language C.

Second, even statements in C can’t run directly on actual hardware. They need to be compiled into instructions that can run on a processor/computer system. We’ll be learning about one of the two most common assembly programming languages used today, ARM. ARM is used in most mobile processors as well as some laptops (newer Apple Macs) and is friendly to those learning how to program in assembly. Although many will not likely program much assembly in your career, you will almost certainly need to know how to read it. Moreover, if you have any interest in more advanced topics in computer systems, computer architecture, or computer security, you’ll need to know assembly language.

Third, assembly instructions translate nearly directly into machine instructions that are executed by computer processors. As a computer scientist or computer engineer you should have a basic understanding of how these instructions execute. This course will give you a preview of digital logic and processor design to set you up for success in CSE 140, CSE 141, and CSE 142.

How, exactly, are you going to accomplish this?

In lecture, you will learn about the concepts underlying computer systems. You’ll then have opportunities to practice and cement this understanding by programming in C and assembly and by completing conceptual homework.

How will you (and the professor) know if you are making progress in your learning?

The course offers you numerous opportunities for feedback on whether you are learning what you need to know. Learning about computing isn’t done by just reading about it, so you’ll be asked to solve problems related to the class in programming projects and homeworks. Beyond learning the topics of this course, I care a lot about you developing your general skills in technical analysis and
communication. So throughout the class, you will be able to get practice and feedback in the following ways:

**Reading quizzes**: Before the lecture you will often complete a reading from the text. You can take these brief online reading quizzes to check that you got the key ideas from the reading.

**Homework and Programming Assignments**: This is your opportunity to see if you understand the course material. Try your best to work through the homework and programming assignments entirely on your own, but feel free to get help from the tutors, TAs, or professor. For programming assignments, we strongly encourage working until you pass the majority of test cases **and to create your own test cases**.

**Discussion Sections**: These are held weekly with a TA. They will review lecture material and homework. You may attend any scheduled discussion section. This is an opportunity to have more discussion than is possible in lecture so bring your questions or doubts. Students who are not demonstrating appropriate mastery of the material (based on homework performance or midterm performance) may be required to attend discussion sections in order to pass the class.

---

**Important Course Details**

**Instructor**: Bryan Chin (he/him/his), Keith Muller (he/him/his)

Office Hours: Please see Canvas and our google class calendar for most accurate dates/times (Instructors and TAs have office hours).

**TAs**: Rasika Bhave, Jinhao Liu, Sananya Majumder, Jerry Yu

**Tutors**: Nitya Agarwal; Alexander G Arias; James Bao; Shubham Bhargava; Adrian Botvinik; Tianyi Irene Chen; Melina Kapsogeorgou; Thuan Quang Do; Ruilin Hu; William Hu; Jinya Jiang; Mihir Kekkar; Austin Li; Binghong Li; Frank Li; Gerui Li; Meihui Liu; Rana Lulla; Saransh Malik; Hyunseo Park; Annie Phan; Prashanth Rajan; Adrian Rosing; Jordan Ruggles; Andrew Russell; Jose Salazar; Arjun Sampath; Michael Shao; Xiyi Shao; Catherine Shen; Marcelo Shen; Christian Sulaiman; Timothy Wu; Benjamin Xia; Zenas Zhu

**Lab hours**: these are for one on one help and are both in person and remote. See autograder.ucsd.edu for lab hour schedule

**Class website**: canvas.ucsd.edu

All Course materials will be on canvas and/or piazza this quarter. Please be sure you have accounts on both canvas.ucsd.edu and piazza for this class (your canvas enrollment should synchronize you with a piazza account in most cases). You are responsible for reading and staying up to date with the class on Canvas. Announcements, assignments, etc. will be posted on this site and on piazza (see piazza site for homework, concept questions). **gradescope** will be used for homework assignments. **Please see piazza for a list of all our class platforms used this quarter.**

**Class lab hours**: Tutors will be available online and in person to help you throughout the quarter.

**Meeting times and places:**

Lecture: You may attend any lecture you’d like
Section A (Muller) : T/TH 12:30 - 1:50 JEANN AUD
Section B (Chin) : MWF 3:00 - 3:50 LEDDN AUD
Section A01 Discussion: M 5-5:50 WLH 2005
Section A02 Discussion : M 6-6:50 WLH 2005
Section B01 Discussion : F 12-12:50 MOS 113

All lectures and discussions are recorded for later viewing at podcast.ucsd.edu.

Lecture Attendance is **HIGHLY RECOMMENDED**.
Exams: (in-person)
Midterm Exam : Thursday Oct 27, 2022 - evening
Final exam: December 9, 2022: Section A - 11:30 - 2:29, Section B 3:00 - 5:59

Course materials

- (Required) Mathews, Newhall, and Webb, “Dive into Systems”
  Available free online
- (Required) Szuhay, Jeff, "Learn C Programming: a beginner's guide to learning C programming
  the easy and disciplined way" - Second Edition
  Available free online from ucsd library

Other recommended reading

- Kernighan and Ritchie, "The C Programming Language 2nd Edition"
- David Griffiths, Dawn Griffiths, Brain Sawyer; Head first C
  Available online thru the UCSD library.
- "Learning the vi and Vim Editors, 8th Edition", Arnold Robbins, Elbert Hannah, O'Reilly Media
- Harris and Harris, "Digital Design & Computer Architecture Arm Edition"
  An online copy can be obtained via the campus library system

Class Announcements
All announcements, updates on homework assignments, etc. will be posted on canvas and/or piazza. All students are responsible for announcements and information on canvas and piazza.

Lecture Notes
We will lecture writing on slides and will release our slides before and after the class as a pdf. We will also record all lectures.

Course Outline:
I. Number Systems
II. C Programming
III. ARM Programming
IV. Digital Logic and CPU Architecture

Grading Information:
All of your class work will total to 1000 course points as follows. Individual assignments points may be scaled as appropriate to the following course points.

- Reading Quizzes 50 pts
- PAs (programming/homework) 9 x 50 : 450 pts
- Midterm: 150 pts
- Final: 350 pts

Reading Quizzes: You are expected to come to class prepared to learn and that means you should do the assigned reading. For most reading assignments there will be an assigned quiz on Canvas for you to test your understanding. In some cases, quizzes will be assigned related to course content, but not necessarily the reading. All quizzes are equally weighted.
Written Homework and Programming Assignments allow you to test your understanding of the material and gain feedback on that understanding. All homeworks and programming assignments are weighted equally (regardless of points allocated) unless stated otherwise. Each homework is nominally worth 50 points (or scaled to 50 points) and all 9 homework/PAs total 450 points. Generally, homework is due on Wednesday one week after it is assigned. Homework turned in more than 3 days after the due date will receive a 0 (see late policy). **Turning in a partial homework is better than receiving no credit.**

- **Programming Assignments and/or written homework** MUST be submitted as directed through gradescope.

- Gradescope will automatically grade part of your code and you’ll get feedback on basic MINIMAL tests. Gradescope will run more advanced tests after the submission deadline – it is your responsibility to thoroughly test your code before submission. **DO NOT RELY SOLELY ON THE PUBLIC TESTS. GRADESCOPE'S AUTOGRADER IS NOT AN ACCEPTABLE TESTING STRATEGY.**

- **Late work:** You can submit Programming Assignments late with a 5 point penalty per day up to 3 days (15 point deduction). **No assignment will be accepted after 3 days.** The 5 point penalty is applied after all slip days have been used at the end of the quarter. **This will not be reflected in the gradescope score.**

- **Slip days:** All students start with a credit of 8 slip days. These are applied at the end of the quarter. The slip days will remove any penalties associated with assignments that are turned in late up to the late submission date. At the end of the quarter, we will add up all your late days and subtract **5 pts x max( late days - 8, 0).** For example, if you hand in all 9 PA two days late each, then your total late days is 18 and we will deduct 50 pts from your homework grade (5 pts x max(18-8, 0)). If you submit PA1 3 days late, PA2 3 days late, PA3 2 days late, and all the rest of the PAs on time, you will receive no late penalty. Any assignment turned in after 3 days will receive a 0 and slip days will not be used. So if you hand in PA1 4 days late, you will receive a 0 on PA1 but not use any slip days.

- If you submit a homework multiple times, the latest submission is what will be graded. However, manually graded items (such as code comments) will only be graded once and we reserve the right to grade the submission that was available at the time of grading.

- Regrade requests must be submitted within 3 days of the grades being released or per assignment policy. Regrade requests must have justification.

- You must work individually on each assignment unless otherwise specified.

- Assignments that do not compile or do not run, or are misnamed, may be given 0 points.

**Exams:**

- **Midterm:** The midterm is worth 150 points.

- **Final Exam:** The final exam is worth 350 points. The final exam may be inclusive of all course material. **You must pass the final exam to pass the class.** 65% or better after a possible curve is applied is considered passing.
Regrade requests. If you feel we made a mistake in grading (for a homework or the midterm), you may submit a regrade request on gradescope. Ask for a regrade if you genuinely believe we have made an error in grading and can justify it. Frivolous regrade requests may result in additional penalties.

Course Grade:
- We will use a standard scale for assigning letter grades: 90-100 = A; 80-89.9 = B, 70-79.9 = C, 60-69.9 = D, <60 = F. Pluses and minuses will be given at the instructor’s discretion. For example, if you score between 90-100, you are guaranteed some kind of A, but whether it's an A+, A, or A- will depend on several things (including a possible course curve, final exam performance, professionalism, and possibly others). The course may be curved at the instructor’s discretion. YOU MUST PASS THE FINAL EXAM TO PASS THE CLASS.
- If you are taking the course pass/fail, you must get at least a C- to pass. If you are a graduate student taking it Sat/Unsat you need a B- to pass.

Special circumstances:
- The only exceptions to the rules regarding late assignments or exams are extended absences (one week or more) due to verifiable extraordinary circumstances, and absences due to official UCSD activity travel. In the case of absences due to a UCSD activity travel, you must give the instructor a list of your travel dates as soon as it is available. In case of extended illness, contact your college Dean's office and work with them to coordinate with us and your other classes.
- If you are eligible for accommodations as per UCSD OSD policies, you must contact the instructor by the end of week 2 to get them arranged.

Integrity:
- Review and sign the course How to Excel with Integrity Agreement (this is homework which will be assigned)
- Cheating WILL be taken seriously. It is not fair to honest students to take cheating lightly, nor is it fair to the cheater to let him/her go on thinking that cheating is a reasonable alternative in life.
- The following are not considered cheating:
  - Working on programming assignments or homework alone
  - Talking over an assignment with one of our course staff (Instructor, TA, Tutor).
  - Using internet resources of a generic nature (e.g. googling “hash tables”)
  - Having a git repository record of incremental work.

The following are examples of forbidden activities:
- discussing in any form (written or oral) the assignment implementation, pseudo code or approach with another person other than course staff before all parties have submitted the assignment.
- copying material to or from the internet (sites such as Chegg, Course Hero)
- copying source code from the internet (e.g. google “hash tables” and then copy the code from the internet)
- showing your code or homework to someone else other than course staff
- publishing your code or homework on the internet such that it is accessible to others with the notable exception of course staff. (e.g. no public github repositories).
- representing work done by others as your own.

Review the Integrity of Scholarship Agreement for more details about what is permitted.

WHEN IN DOUBT, ASK US!!!!
- To detect instances of academic integrity violations in programming assignments we will use 3rd party software. We recommend you only include your class lab account ID (not your name or PID) in your submissions. Including your name and/or PID may disclose that information to the 3rd party.
Homeworks and projects are not intended to be the grade-makers, but to prepare you for the tests, which are the grade-makers. Cheating on a homework or project is not only unethical, but shows a fundamental misunderstanding of the purpose of homework and projects.

Penalties -- If we become aware of any violations of these rules by a student we are obligated by Academic Senate policy to initiate the actions described in the Policy on Academic Integrity. Integrity violations may result in a zero for the assignment, a zero for that portion of your grade, or an “F” in the course, among others.

IN THE PAST - WE HAVE HAD STUDENTS FACE ACADEMIC SANCTIONS DUE TO ACADEMIC INTEGRITY VIOLATIONS IN THIS COURSE. These have varied from an F in this class to academic suspension or expulsion from the University. DO NOT LET THIS HAPPEN TO YOU.

Getting Help
For this course- you should always consider asking the staff through Piazza, coming to office hours, or scheduling an appointment. **Don't suffer in silence. Be Proactive. We want to help you and there is no shame in asking for help.** We try to answer questions on Piazza in a timely manner (within one day, probably less). Do not expect response to piazza posts within minutes from the staff.

Other kinds of help:
“The IDEA Engineering Student Center, located just off the lobby of Jacobs Hall, is a hub for student engagement, academic enrichment, personal/professional development, leadership, community involvement, and a respectful learning environment for all. The Center offers a variety of programs, listed in the IDEA Center Facebook page at [http://www.facebook.com/ucsdidea/](http://www.facebook.com/ucsdidea/) (you are welcome to Like this page!) and the Center web site at [http://idea.ucsd.edu/](http://idea.ucsd.edu/). The IDEA Center programs support both undergraduate students and graduate students.”

Diversity and Inclusion
We are committed to fostering a learning environment for this course that supports a diversity of thoughts, perspectives and experiences, and respects your identities (including race, ethnicity, heritage, gender, sex, class, sexuality, religion, ability, age, educational background, etc.). Our goal is to create a diverse and inclusive learning environment where all students feel comfortable and can thrive.

Our instructional staff will make a concerted effort to be welcoming and inclusive to the wide diversity of students in this course. If there is a way we can make you feel more included please let one of the course staff know, either in person (or on zoom), via email/discussion board, or even in a note under the door (or our e-mail inboxes). Our learning about diverse perspectives and identities is an ongoing process, and we welcome your perspectives and input.

We also expect that you, as a student in this course, will honor and respect your classmates, abiding by the UCSD Principles of Community ([https://ucsd.edu/about/principles.html](https://ucsd.edu/about/principles.html)). Please understand that others’ backgrounds, perspectives and experiences may be different from your own, and help us to build an environment where everyone is respected and feels comfortable.

If you experience any sort of harassment or discrimination, please contact the instructor or course staff as soon as possible. If you prefer to speak with someone outside of the course, please contact the Office of Prevention of Harassment and Discrimination: [https://ophd.ucsd.edu/](https://ophd.ucsd.edu/).

Basic Needs
If you find yourself hungry or unable to meet your basic nutritional needs or suffer from housing insecurity or other basic needs issues (or know someone like this): [http://basicneeds.ucsd.edu](http://basicneeds.ucsd.edu)

Triton Food Pantry is free and anonymous, Financial aid resources are available. CAPS and college deans can help connect to the above resources as well as other support.