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) and designing more advanced programs (CSE 12). 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 to you. 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 top two assembly programming languages used today, ARM. ARM is used in most mobile processors today and is friendly to those learning how to program in assembly. Although you will likely not program much in 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 very well.

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. You 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:

· **(Optional) Reading quizzes**: Before 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 homeworks, be sure to review solutions to identify areas of strengths and weaknesses. For programming assignments, we strongly encourage working until you pass the majority of test cases. Even if you don’t complete the programming assignment in time, we’ll give you an opportunity to keep learning by testing your code in an assignment on gradescope (that does not impact your grade).

· **Peer Instruction Questions**: For those of you who attend Prof. Porter’s lectures, you will have an opportunity to deepen your understanding of computer organization concepts while developing technical communication skills by discussing challenging questions in a team of three to four students. Feel free to ask questions of TAs, tutors, and/or the professor during the discussion time.

We encourage you to make the most of this feedback in order to ensure that you perform well on the course quizzes and exams!

**Important Course Details**

**Instructors**: Keith Muller and Leo Porter
- **OH**: Please see course website for most accurate dates/times.
  (We’re happy to add more if this is not sufficient)

**TAs**: Erika Auyeung, Rachel Chung, Ranga Ramkumar, and Richa Pallavi

**Tutors**: There are 12 tutors for the course. Please see canvas/piazza for details.

**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. You are responsible for reading and staying up to date with the class website. Announcements, assignments, etc. will be posted on this site and on piazza (see piazza site for homework, concept questions). **gradescope** will be used for programming assignment and homework submission. **Please see piazza for a list of all our class platforms used this quarter.**

**Class open lab**: Tutors will be available online to help you throughout the quarter.

**Meeting times and places**:
- **Lecture**: You may attend any lecture you’d like (capped at 200 per section)
  - Section A: Tu/Th 9:30-10:50am on Zoom -- interactive format with questions/discussion
  - Section B/C: Tu/Th 11:00-12:20pm on Zoom -- interactive format with questions/discussion
  - Section D: Tu/Th 12:30-1:50pm on Zoom -- focus is content delivery, but you can still ask questions
- **Discussion**: In lieu of traditional discussion, you will be expected to sign up for weekly 30min-1 hour meetings with roughly 10 students and 1 TA/Tutor to go over your questions and work through some sample problems.
- There will be four quizzes to be held roughly every other week. See out tentative schedule for details.
- **Final exam**: All sections will have a time window on Saturday 12/12/2020 to take the exam.

**Course materials (required)**
- Harris and Harris, "Digital Design & Computer Architecture Arm Edition"
  - An online copy can be obtained via the campus library system
- Mathews, Newhall, and Webb, “Dive into Systems”
  - An access code to the online textbook will be provided on canvas
Other recommended reading
○ Kernighan an Ritchie, "The C Programming Language"

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 ppt slides and will release our final slides as pdf after class. We will also record all lectures.

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

Grading Information:
○ The grade for this class (out of 100% total) will be as follows:
  ● Online Quizzes: 2%
  ● Professionalism: 1%
  ● Homework and Programming Assignments: 30%
  ● Quizzes: 33%
  ● Final: 34%
○ (Optional) Reading quizzes: You are expected to come to class prepared to learn and that means you should do the assigned reading and complete the assigned reading quiz. In some cases, quizzes will be assigned related to course content, but not necessarily reading quizzes. The combination of coming to class prepared and participating in class (see below) should enable you to excel throughout the class.
○ Online quizzes: At various points in the class we’ll need you to turn in feedback or short answers regarding class content. These will be clearly announced on piazza.
○ (Optional) Class Participation: You are highly encouraged to attend class and participate in questions and discussion but attendance will not be taken.
○ (Partially Optional) Small Group Discussion Participation: Weekly, we expect all students to join with TAs/Tutors in small groups to talk about the class and to work through practice problems. Although this is highly encouraged, it is not required. However, see the Quiz description below to see the benefit of participating.
○ Professionalism: We expect everyone in the class to conduct themselves in a professional manner. We consider professional behavior to be a program-level academic outcome of UCSD. Though broadly defined here, it includes (at least), professional conduct with the teaching staff and your fellow classmates. Some examples of unprofessionalism in prior courses include: excessive arguing with teaching staff over assessment outcomes (grades), belittling/rude/unkind behavior toward other students or teaching staff, and excessive lack of resourcefulness (e.g., e-mailing course staff with questions already answered on piazza or in this syllabus). Though rare, we reserve the right to deduct participation points from repeated unprofessionalism. Note that particularly severe infractions (e.g., sexism, racism, lying - which are never tolerated in our community) may be subject to campus Academic Conduct Charges.
○ Homework: Homeworks allow you test your understanding of the material and gain feedback on that understanding.
  ● To facilitate quick feedback, homework MUST be submitted as directed (likely through gradescope). It must be submitted on-time through the platform. Allow yourself time for unexpected platform hiccups and be sure to follow instructions.
  ● All homework is to be done individually. You may ask questions about the homework on piazza and you may speak with the TA or Professor in OH. See below regarding acceptable, minimal student collaboration.
• **Late homework assignments are not allowed.** We will make every effort to return assignments to you and post solutions in a timely manner -- limiting your ability to turn things in late is, unfortunately, critical to that goal. **You may drop your lowest homework OR programming assignment grade – use that dropped score judiciously!**
  • You may use gradescope regrades to identify grading errors on homeworks. ONLY fully correct solutions which are incorrectly marked are eligible for regrading (do not use regrades to argue for additional partial credit). Regrades must be submitted within 3 days of the grades being released.
  • You are highly encouraged to discuss homework problems with the professor, TAs, and tutors in office hours. If you do not understand a concept, it pays to address it as soon as possible.
  ○ **Programming Assignments**
    • You must submit your assignment following the directions provided with the assignment (either through ieng6 or through gradescope).
    • Gradescope will automatically grade part of your code and you’ll get feedback on those basic tests. Gradescope will run more advanced tests after the deadline – **it is your responsibility to thoroughly test your code before submission.**
      • Regrades must be submitted within 3 days of the grades being released.
      • You must work individually on each assignment.
      • Assignments that are submitted late, do not compile or do not run, or are misnamed, will be given 0 points. **You may drop your lowest homework OR programming assignment grade – use that dropped score judiciously!**
  ○ **Exams:**
    • **Quizzes:** There will be quizzes roughly every other week. You can drop your lowest quiz score. In addition, you can earn back 25% of your lost points by attending and participating in both small group discussions over the two weeks following the quiz. For example, if you receive a 60% on Quiz 1 held at the end of week 2, if you attend and participate in your assigned small group discussion in Weeks 3 and 4, your grade will be updated to .6 + (.25*(1-.6)) = .7. Participation in the discussion means that you answer the TA/Tutor when called on and/or you ask questions during the session. You must be on time and present for the whole section to count as “attending”.
    • **Final exam:** The final will be inclusive of all course material.
  ○ **There are no alternative/make-up exams.**
    • You have the right of appeal for grading on all tests. Please check your exam over carefully when you receive it and be sure to request a regrade in the case of a grading error. Also review your exam to understand where you made errors, if in doubt, come to see us in Office Hours (do not use regrade requests to find out what you did wrong). Note that frivolous regrade requests may result in additional points deducted.
      • 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.
      • 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, and possibly others). The course may be curved at the instructor’s discretion.
      • The only exceptions to the rules regarding no 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.
      • 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 organized.

**Integrity:**
  • **Review and sign** the course Integrity of Scholarship Agreement
  • 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 is not considered cheating:
      • Working on programming assignments or homework alone
      • Working in a group to discuss an assignment with the following restrictions:
1. Nothing during that discussion can be taken away from the discussion other than your brain (i.e. no pictures of notes written on the whiteboard, no writing down what is discussed)
2. Any code discussed remains pseudocode
   - The following is considered cheating:
     * using homework or programming solutions from the web, previous versions of the class, your friends, or anywhere else.
     * receiving, providing, or soliciting assistance from another person/device during a test.
   - Review the Integrity of Scholarship Agreement for full details about what is permitted.
   - 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, an “F” in the course, among others.

Getting Help
For this course- you should always consider asking the staff thru 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.

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/ (you are welcome to Like this page!) and the Center website at 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, via email/discussion board, or even in a note under the door. 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). Please understand that others’ backgrounds, perspectives and experiences may be different than 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 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/
No Tolerance for Harassment and Discrimination

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
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.