University of California, San Diego

CSE 30: Computer Organization and Systems Programming
Spring 2022 (Version 1.10)

Introduction

You've been writing programs for the past year (or more), you have become more capable of analyzing the efficiency of programs (CSE 12 and 21), working in the Linux environment (CSE 15L), and designing more advanced programs (CSE 12). However, the computer likely remains nearly a black box for you.

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 high-level programming language semantics translate into low-level programming (assembly language). Explore the basics of the underlying hardware (gates, memory) and how high-level programming language semantics are implemented and how they influence the basic architecture of a CPU and the design of an operating system.

The goal is to give you some insight into and 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.

Course Description

This course is designed to convey three critical ideas in computing to you. First, you've spent most of your time programming in Java or Python. While Java (or Python) makes programming easier for you, it hides away important details of how programs really work, how they are implemented and how they interact with the operating system (Linux specifically). 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 one of the most widespread assembly programming languages in use today, ARM (Aarch32). Variations of ARM assembly are used in most mobile processors today as well as some computers (latest Apple M1 chip). ARM assembly is a good start to learning how assembly language works as it is a relatively simple design. Although you will likely not program a lot of assembly in your career, you will almost certainly need to know how to read it when debugging programs and understanding how certain high-level programming language semantics impact program execution performance and memory use. Moreover, if you have any interest in more advanced topics in computer systems, such as operating 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.

In lecture, you will learn about the concepts underlying computer systems. You will have opportunities to practice and cement this understanding by programming in C and assembly and by practice with conceptual homework. The course will focus on C and ARM (32-bit) assembly languages implementations using the Linux operating system running on the Raspberry PI4.

Prerequisite

CSE 12 and CSE 15L. It is expected you have experience in either JAVA, Python, or C. We will be teaching the specifics of C in class under the assumption that you have already have some programming experience, so this will not be an introductory level class.

Learning Outcomes

At the completion of this course, you will have acquired a strong competency in programming in C and ARM assembly language (individually and combined in a single program) in the Linux runtime environment.

You will understand how high-level language semantics are implemented in assembly language and how they influence and interact with the design and implementation of the operating system (Linux) and the CPU Instruction Set Architecture. Understanding the low-level runtime environment and implementation is an invaluable asset in debugging programs for proper execution, security, portability and optimizing runtime performance.

You will learn how subroutines are implemented (both in runtime software and in assembly language instructions), how local and global variables are allocated memory space, how stack frames work, how to pass parameters in registers and on the stack, how to use pointers, how to combine assembly language functions and C functions in a program, and how to use trap instructions to call operating system calls.

You will learn how programming languages are converted into machine code and how machine code is executed by a CPU (using a basic single cycle CPU). We will investigate the basic hardware building blocks (memory, mux, ALU, registers), and how they are combined into creating a CPU.

We will look at the representation of data in memory and what are the implications of this to the programmer and the CPU Instruction Set Architecture.

Course Logistics

Instructor and Course Staff

Office Hours: Please see Canvas and the class Google Calendar for most accurate date/times.

Instructor: Keith Muller (muller at eng.ucsd.edu)

TA's: (Leads Erika Hoi-Kay Auyeung, Soham Satyadharma), Ziyu Lin, Sananya Majumder, Srirangan Madhavan

Tutors: Nitya Agarwal, Mudit Bajaj, Adrian Alex Botvinik, Victor Wei Chen, Daniel Doan, Ruilin Hu, Michael Ivanov, Yuchen Jing, Mihir Sanjay Kekkar, Jason Yongjae Lee, Binghong Li, Frank Li, Gerui Li, Jinhao Liu, Rana Lulla, Samuel K Ng, Hyunseo Park, Prashanth Rajan, Arjun Dilip Sampath, Michael Liu Shao, Alex Simonyan, Christian Sulaiman, Shijie Wang, Timothy Zhong Zun Wu, Warren Xia, Alexander William Yu

Class website: https://sites.google.com/ucsd.edu/cse30spring22
All Course materials will be on canvas and/or Edstem this quarter. Please be sure you have accounts on both canvas.ucsd.edu and Edstem for this class. You are responsible for reading and staying up to date with the class on both Canvas and Edstem. Announcements, assignments, etc. will be posted on canvas and on Edstem (see **Edstem** site for homework, concept questions). **Gradescope** will be used for homework assignments.

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

Course Resources

The following textbooks are available free online at the campus library:

"Head First C", David Griffiths, Dawn Griffiths; (Required)

https://search-library.ucsd.edu/permalink/01UCS_SDI/ld412s/alma9914828943806531

"C in a Nutshell, 2nd Edition", Tony Crawford, Peter Printz; (secondary/alternative)

https://search-library.ucsd.edu/permalink/01UCS_SDI/1vtf07t/cdi_safari_books_v2_9781491924174

"Understanding and Using C Pointers", Richard Reese; (sections)

https://search-library.ucsd.edu/permalink/01UCS_SDI/1vtf07t/cdi_askewsholts_vlebooks_9781449344559

"Digital Design and Computer Architecture: Arm Edition", Harris and Harris (sections)
https://search-library.ucsd.edu/permalink/01UCS_SDI/ld412s/alma991008932009706535

"Art of Debugging with GDB, DDD, and Eclipse", Norma Matloff, Peter Salzman (reference only)
https://search-library.ucsd.edu/permalink/01UCS_SDI/ld412s/alma991007441759706535

"Learning vi & vim Editors, 8th Edition", Arnold Robbins, Elbert Hannah (reference only)
https://search-library.ucsd.edu/permalink/01UCS_SDI/1vtf07t/cdi_safari_books_v2_9781492078791

The following textbooks are available free on the web:

"Introduction to Computer Organization: Arm Assembly Language", Platz (sections)
https://bob.cs.sonoma.edu/IntroCompOrg-RPi/intro-co-rpi.html

"Dive Into Systems", Mathews, Newhall, Webb; (sections)
https://diveintosystems.org/book/

Edstem Discussion Board – Q/A, Announcements, and Discussions

Always post here with your questions instead of emailing staff for help!

1. If you have a question that you can express without revealing any of your assignment solution, then you may post it to the course's Ed Discussion forum. One of the course's staff will respond as soon as possible. Other students are also welcome to respond if they can do so without revealing any of their *assignment solutions* either.
2. Study provided material before posting questions including Lecture slides, discussion handouts, textbook readings, class videos, etc.
3. Read / search all (recent) Edstem threads before posting question
4. **Remember:** **Never** reveal your *assignment solution* in a public post!!!! Always Click “private” when making an Edstem post if in doubt.

*"Assignment solution (for a grade)" **in the context of PUBLIC edstem postings** includes*:

1. The products that you create for the assignment (for a grade), that is, your source code files (including comments), basically any file or modified versions or parts of them you turn in for a grade, etc. You may not show any of the products that you create for the assignment, modified versions of them, or any parts of them, including any test files you create on Edstem in a public post.
2. Descriptions of those products in the form of pseudo-code, flow charts, outlines, diagrams, natural language prose, etc. You may not describe the products that you create for the assignment (for a grade), or any parts of them, on Edstem in a public post.

Violations on public Edstem postings are considered an AI violation.

Time and Location

Lecture: (See Canvas/Edstem for ZOOM ID and recorded)

Monday, Wednesday, Friday 1:00 PM – 1:50 PM; Jeannie Auditorium

– Required –

You are strongly encouraged to come to lecture (in person) and participate by asking questions. ***You can also attend this class live via zoom (asking questions via chat) or watch the recorded zoom lectures offline.*** Please keep up as binge watching the lectures in a few hours has caused a lot of students in prior quarters to get behind and to the detriment of their grades.

Why is lecture attendance is required: Some of the material covered in lecture is not available in any of the textbooks. All material covered in lecture may be tested for in quizzes and the final as well as being necessary to complete the programming assignments. **This is extremely important for the ARM programming part of the class** where CSE 30 focuses on the 32-bit ARMv6 version (not thumb or thumb-2) of assembly language under Linux on raspberry pi4 hardware.

All material and guidance presented in lecture takes precedence over the material found in the textbooks and other web resources.

Please download and review the posted lecture pptx slides BEFORE lecture and do the assigned readings by the due date.

We encourage you to listen attentively to the lectures, but also participate actively by asking helpful questions. When attending in-person, ask questions to the instructor. On Zoom (live) post questions to chat (a TA will answer them, or the TA will ask the instructor for you).

Final Exam: Required (In-person)

Thursday June 9 11:30 AM – 2:29 PM; Jeannie Auditorium

You MUST contact the instructor via e-mail before April 1 @ 5:00 PM if the scheduled final date/time as shown in the schedule of classes (or the in-person requirement) does not work for you, to see if an alternative can be agreed on. ***Please, only use e-mail communication to request alternative arrangements.***

Discussion Sections: (see Canvas/Edstem for ZOOM ID) – Optional but strongly encouraged

Section A: Monday 4:00 PM – 4:50 PM; Solis 107
Section B: Monday 5:00 PM – 5:50 PM; Solis 107

Quizzes and Programming Assignments

There are four (4) online quizzes. There are five (5) programming assignments distributed over the quarter.

Weekly Reading Assignments and Practice Questions

Weekly reading assignments will be posted on canvas that reference the course textbooks. Along with the reading assignments there will be some ungraded practice problems (with answers). These will be reviewed in optional weekly zoom videos and questions may be asked in the discussion sections

Class Announcements

All announcements, updates on homework assignments, etc. will be posted on canvas and/or Edstem. All students are responsible for announcements and information on both Canvas and Edstem. Please check both at least daily.

Lecture Slides

Slides are available for download in PowerPoint format (.pptx) from canvas about 1 week before lecture. All lectures are available via ZOOM and the recording can be viewed on canvas.

Discussion Session Slides

Slides are available for download before the discussion section. All discussion sections are available via ZOOM and the recording can be viewed on canvas.

Special Topic Videos and Zoom Sessions (Optional)

During the quarter, lecture and discussion sessions will be supplemented with occasional videos that go over lecture topics to provide more examples than can be accomplished in lecture. Additional review material from prior courses, as it pertains to CSE 30 may also be presented. Practice examples of using the various development tools may also be given in the videos. Review for weekly reading practice problems. Review material for quizzes and the final exam will be in videos and/or live zoom sessions.

Coursework and Grades

Coursework

Course Outline:

I. C Programming
II. Number Systems and Data Encoding
III. ARM (32-bit) Programming
IV. Basics of Digital Logic and CPU Architecture

You are responsible for all the material from the following sources:

1. Contained in Lecture Slides
2. Contained in Assigned readings

3. Ungraded Practice Questions (Solutions provided)
4. Experience acquired by completing all the Programming Assignments

Grading Policy

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.

Important: Regarding Ungraded PA0

As part of PA0 you must read this syllabus and sign that you have read and agree to abide by the policies of this class as described in this syllabus. Your agreement will be submitted as an ungraded PA0 to canvas. **You will not be allowed to take quizzes or turn in programming assignments until this form is signed and submitted.**

Final Exam – 28% of the points

Inclusive of all material in the course

AI Infraction: F in course

Quizzes – 12% total of the points

Four (4) Quizzes: 3% of the points each
On Canvas - ~45 minutes –
Open 12:01 AM; due at 11:59 PM UCSD Time
Due Tuesday week: 3, 5, 8, 10
AI Infraction: 0% on the quiz

Programming Assignments – 60% of the points

Five (5) Assignments: 12% of the points each
Due week: 2 (Thur), 4 (Thur), 6 (Tue), 7 (Thur), 9 (Thur)
Due at 11:59 PM UCSD time
PA released week: 1 (Mon), 3 (Mon), 4 (Mon), 6 (Mon), 7 (Mon)
AI Infraction: Up to 0% on the assignment – at instructions discretion

Programming Assignment Point Distribution Tiers

Tier 0: Code style and comments
Tier 1: Minimum Viable Product Tests

- These are public tests, some may be supplied with assignment

Tier 2: The Field Trial Tests

- These are tests only available on Gradescope

Tier 3: Best in Class Award (not all assignments): bonus points

1. Passing special tests and/or achieving performance metrics as described in the assignment will get this bonus
2. Only eligible with a 100% pass on all Tier 1 and Tier 2 tests
3. PA Must be submitted on time (no late day waivers)

Final 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 an 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).

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 Satisfactory/Unsatisfactory you will need a B- to pass.

The final grade, the final exam, any PA, or any quiz in the course may be curved up at the instructor's discretion.

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 (including but not limited to Edstem points), and excessive lack of resourcefulness (e.g., e-mailing course staff with questions already answered on Edstem or in this syllabus). Repeated occurrences of class disruptions (taking phone calls during class, using technology device displays that contain material not related to the class, etc.) will be considered unprofessional behavior.

Though rare, *we reserve the right to penalize up to 5% of a students' overall grade for repeated acts of 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 and penalties beyond professionalism penalties.

Course Policies

Programming Assignment Policy

1. **Must be submitted only through Gradescope. All other form of submissions without the instructor consent (e-mail or written) will not be accepted!**
2. **Late assignments, 25% off each day –** no exceptions
   >= 1 minute late after due date is considered one day late
3. **Up to 5 late day waivers, max 3 per PA – firm limit, no exceptions**
   Example: you turn in all 5 PAs, 4 are 1 day late and 1 is 2 days late
   Four (4) PAs would have no late penalty
   One (1) PA would have a one-day late penalty
4. **Late day waivers** are applied ***automatically*** at the **end of the quarter** when grades are calculated
5. **Additional late day waivers are available only in the case of extreme unforeseen circumstances and only when documented by your College Dean's office.**
6. Be careful about late submissions. If you had buggy code that was getting an 80% and turn in code a day late (using no waivers) and get a 90% but one day late, the final grade on the assignment would 0.85 X 90 or a 76.5%
7. **No dropped Programming assignments – firm limit, no exceptions**
   Doing the assignment is important to learn & practice the material

Quiz Policy

1. **Are taken only on Canvas. All other form of submissions without the instructor consent (e-mail or written) will not be accepted!**
2. **Taking the quiz late, 50% off each day –** no exceptions
   >= 1 minute late after due date is considered one day late
3. **Up to 3 late day waivers, max 1 per quiz – firm limit, no exceptions**

Example: you take all 4 quizzes, each one is 1 day late
Three (3) quizzes would have no late penalty
One (1) quiz would have a late penalty

4. **Late day waivers** are applied ***automatically*** at the **end of the quarter** when grades are calculated
5. **Additional late day waivers are available only in the case of extreme unforeseen circumstances and only when documented by your College Dean's office.**
6. You may only take each quiz once.
7. **No dropped quizzes – firm limit, no exceptions**
   Taking the quiz is important to learn & practice the material

Final Exam Policy

1. Missing the final exam will result in a score of 0% on the exam.
2. You may take the final exam only once.
3. In the case of extreme unforeseen circumstances, you must contact the ***instructor via email only*** as soon as possible. The instructor will require formal documentation by your Colleges Dean office.
4. **You MUST contact the instructor via e-mail before April 1 @ 5:00 PM if the scheduled final date/time as shown in the schedule of classes (or the in-person requirement) does not work for you, to see if an alternative can be agreed on. *Please, only use e-mail communication to request alternative arrangements.***

Regrade Policy

You have the right to appeal any grade with the instructor, and the final decision is with the instructor only. Regrade appeals for any assignment or exam must be submitted to the instructor via e-mail only, and **within one week (7 days) from when the grade was available to the student**. The student has the responsibility to promptly review graded material and file regrade requests by the deadline. The instructor reserves the right to refuse or accept regrade requests after this time.

Technology Policy

Laptops and other electronic devices like tablets etc. may be used to follow lecture in class. If you must make or receive a phone call during lecture (in-person) please leave the classroom quietly to not disrupt the class. Performing any other tasks on your electronic devices that create sounds or have output on a screen that may be observed by other students in class will distract others and must **not be done** during lecture.

Repeated violation is considered a form of unprofessional behavior (see the section above on unprofessional behavior).

Academic Integrity Policy

In this course we expect students to adhere to the UC San Diego Integrity of Scholarship Policy. This means that you will complete your work honestly, with integrity, and support an environment of integrity within the class for which you are taking. If the course staff suspects that you have violated course policies concerning assignment conduct, then the course staff will investigate, decide, and apply a penalty.

To detect instances of academic integrity violations in programming assignments we will aggressively 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.

The following are additional examples not listed in the General Catalog specific to this course. If you need further clarification, please contact the instructor via e-mail. The term **"assignment solutions for a grade",** is defined to be any assignment including quizzes/exams, etc. that will be (but has not yet) been submitted, graded (and the grade is available) and counts towards a grade in the course.

1. No student shall provide "assignment solutions for a grade", in part or in total in any form (verbally, e-mail messages, pen/pencil on paper, marker on white board, chalk on black board, etc.) to any other student in current or future classes of this course before the other student has submitted and received a grade for that work. This policy holds even after the due date of the assignment for a grade and even after the course is finished.
2. No student shall procure or accept "assignment solutions for a grade" from any other student from current or prior classes of this course, or any other person or source who possesses answers to these or similar assignments and use this in their assignment solutions that will be submitted to be graded. The section of this syllabus pertaining to the use of Edstem provides details concerning the proper use of the course's Edstem forum.
3. Students shall not use any part of an "assignment solutions for a grade" from a previous version of this course or in another or previous course at this or any other university, school, or forum, when completed by another student, as part of their work that they will submit for a grade.
4. No student shall obtain or provide code in part or in whole from any source in an "assignment solutions for a grade" except for code explicitly provided by the instructor or the TAs assigned to the course. Examples of acceptable sources are (1) Canvas or Edstem postings from course staff, (2) lecture or discussion content including contents of course slides not used during class, all class handouts, (3) directly from the instructor or the TAs assigned to this course (see the list of course TA's above**)** in any form (slides, sample code, e-mail, office hours, zoom meetings chat).

5. **No student shall use code found on any internet site other than the course sites in** "assignment solutions for a grade" **unless explicitly authorized only by the instructor.**
6. **No student shall employ another person to help complete or correct their** "assignment solutions for a grade" **before they are graded and returned.**
7. All programming code and documentation submitted for evaluation or existing inside the student’s computer accounts must be the student’s original work or material specifically authorized by the instructor or class policies.
8. Collaborating with other students to develop, complete or correct coursework that is "assignment solutions for a grade" is limited to only those activities explicitly authorized by the instructor. The use of another student’s coursework that will be, in part or in total, to develop, complete or correct course "assignment solutions for a grade" including documentation, **for work not yet submitted and graded** is unauthorized.
9. Students may not engage in pair programming or group work in this course unless explicitly authorized by the instructor. Request authorization via e-mail to the instructor.
10. Use of texting or messaging services or any Internet sites such as Pastebin, discord, CourseHero, Chegg, gopeer.org, or GitHub or similar systems with the intent to share or publish any course files (including Programming Assignment descriptions) or to solicit help not explicitly authorized by these policies, in part or in total to others is unauthorized. Unless otherwise explicitly authorized by the instructor, each student is completely responsible to keep their code, homework, design files and other coursework private on Internet sites. If you are not sure how to keep all access private, ask for help or do not use these internet resources for course work that will be graded.
11. No student shall discuss the content or answers of any exam or quiz with any other student enrolled in any section of this course who has not yet taken the exam until after all the exams are graded and returned. This policy applies to both in-class exams and online quizzes.
12. Examining and/or copying someone else's "assignment solutions for a grade" for use in your "assignment solutions for a grade" is a violation of course Academic Integrity policies.
13. Copying and transforming someone else "assignment solutions for a grade" (by rearranging code, renaming variables, rewording comments, and any other methods employed to obfuscate that your code is derived from another's work) for use in your "assignment solutions for a grade" is a violation of course Academic Integrity policies.

AI Policy Examples: Receiving Help that You May Use

To complete "assignment solutions for a grade" **you may use** the following:

1. All forms of Lecture and discussion section material and handouts
2. **Any material found in Textbooks and any other references listed in this syllabus unless explicitly stated otherwise by the class instructor.**

3. Postings on Edstem from staff or from students that do not violate the Edstem restrictions on posting assignment solutions
4. man pages on the pi cluster (and other UCSD systems)
5. Communication with Staff members during office hours, lab hours, or e-mail.
6. Consult with other people including students that have already taken the course if and only if the interactions do not violate any other Academic Integrity policies. It is your responsibility to make sure the person who is helping you knows the course policies and does not violate them.

AI Policy Examples: Receiving Help that you May Not Use

To get help with "assignment solutions for a grade" you **may not use** the following:

1. Anyone **other than** current course staff members
2. Any student who is currently taking the course or who took the course at a prior time in any form (verbally, computer files, e-mail messages, pen/pencil on paper, marker on white board etc.) at any level of detail (code, pseudo-code, flow charts, outlines, diagrams, natural languages, etc.) at any time (before or while working on your assignment solution for a grade).
3. "Assignment solutions for a grade" before your work is graded (regardless of completeness or correctness) provided by any student who is currently taking the course, provided by any student that has previously taken the course, by any third party, or retrieved from the web, or any other resource system or human. Even glancing momentarily at an assignment solution other than your own – beyond the time required to recognize that it is related to the assignment is a violation of the course academic integrity policies.
4. Material found in textbooks and other internet resources that is copied in part or whole and used in your work in any "assignment solutions for a grade".

AI Policy Examples: Providing Help to Others

1. You may not provide help to another student to complete any class work that has not yet been submitted for a grade. Programming assignments, taking, or answering questions on a quiz or exam in any form (verbally, computer files, e-mail messages, pen/pencil on paper, marker on white board etc.) at any level of detail (code, pseudo-code, flow charts, outlines, diagrams, natural languages, etc.) at any time before or while working on your "assignment solutions for a grade".
2. This policy holds even after the due date/time of the assignment, and even after the course is finished. That is, you may not provide assignment-related ("assignment solutions for a grade") help to students who take the course in future quarters.
3. While taking this class you may provide "assignment solutions for a grade" help to other students on Edstem as long as you follow the Edstem requirements listed in the section on Edstem.

4. After successfully passing this course, you may provide "assignment solutions for a grade" help if acting officially as a TA or tutor for the class following the restrictions that those roles allow.
5. Sharing your work on an "assignment solutions for a grade" with another student before the other students "assignment solutions for a grade" has been graded is an Academic Integrity Violation.

Collaboration Policy

1. Collaborating with other students in groups (two or more) for studying and reviewing class material (textbook readings, lectures (videos and slides), discussion sections (videos and slides) is explicitly allowed and encouraged.
2. Study groups working together to complete ungraded reading assignments are also explicitly allowed and encouraged. Study groups for quizzes and the final exam **before all members of the group have taken the** quiz or exam are also allowed and encouraged.
3. Collaborating with other students in groups (two or more) to complete "assignment solutions for a grade" **before it has been graded for all members of the group** is explicitly prohibited. Be careful: It is acceptable to discuss the requirements of an assignment to refine what the assignment is asking you to do, **but you must not talk about the solution to the "assignment for a grade", before the assignment has been graded for all members of the group, as that is prohibited**.
4. Collaborating with other students in groups (two or more) for studying and reviewing "assignment solutions for a grade" **AFTER it has been graded for all members of the group** is allowed and encouraged.

Outside Tutoring Policy

Individuals are not permitted to approach students to offer services of any kind in exchange for pay, including tutoring services. This is considered solicitation for business and is strictly prohibited by University policy.

Class Policy Waivers

The only person who can grant exceptions or changes from the above stated policies is the CSE 30's instructor-of-record. Waivers from the instructor-of-record will only be granted via e-mail or written communication; claims of verbal waivers are not binding. No other waiver from other sources can modify these policies (unless explicitly stated in these policies or by formal University, School of Engineering, or CSE department rules or policies). It is the student's responsibility to adhere to these policies, and to ignore any implied or explicit waiver from any unauthorized source. For example, if the student asks a TA, Tutor, or friend if something is permissible, any response that deviates from course policy cannot be used to justify any action that is contrary to course policy stated above.

Getting 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 web site 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 are any actions that we can take to help 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/.

Students with Disabilities

We aim to create an environment in which all students can succeed in this course. If you have a disability, please contact the Office for Students with Disability (OSD), which is in University Center 202 behind Center Hall, to discuss appropriate accommodations right away. We will work to provide you with the accommodations you need, but you must first provide a current Authorization for Accommodation (AFA) letter issued by the OSD. You are required to present their AFA letters to Faculty (please decide to contact the instructor privately) and to the OSD Liaison in the department in advance so that accommodations may be arranged.

Basic Needs/Food Insecurities

If you are experiencing any basic needs or insecurities (food, housing, financial resources), there are resources available on campus to help, including The Hub and the Triton Food Pantry. Please visit http://thehub.ucsd.edu/ for more information.