Syllabus
CSE-31-01, Fall 2021

Designation:

CSE 31 Computer Organization

Catalog Description:

Exposes students to the underlying structure of machines. Starting from C programming, pointers, data representation, MIPS instruction-set, Compilation process and down to Hardware implementation.

Textbooks and Other Required Materials:

Main textbook:

Computer Organization and Design from zyBooks
1. Sign in or create an account at learn.zybooks.com using your UC Merced email address
2. Enter zyBook code UCMERCEDCSE031ChandrasekharFall2021
3. Subscribe

Each student must subscribe his/her own copy with their UC MERCED EMAIL ADDRESS. Participation grade will be evaluated based on the activities within the subscription account.

If you encounter a problem, contact support@zybooks.com to get it resolved.

Important Note -- Participation grade will be partly evaluated based on the activities within the subscription account. If you are retaking CSE 31, you may qualify for a retake subscription pricing, and will need to contact support@zybooks.com to avail this subscription.

Supplementary Textbook:

Brian W. Kernighan and Dennis Ritchie. C Programming Language, 2nd Edition, Pearson, 1988. ISBN-13: 978-0131103627.

Course Objectives/Student Learning Outcomes:

Students will learn all the big ideas underpinning the design of modern computers starting from binary numbers representing integers, floating points, data structures and even program code. They will be exposed to C programming language and how each high-level construct translates to intermediate language called Assembly before finally getting down to machine language that modern computers executes. Exact steps required for execution instructions in the CPU will be detailed along with the necessary devices starting from CMOS transistors to gate-level logic using Finite State Machines.

By the end of the course, students will be able to:
1. Describe how data are represented in the computer, including floating point numbers and arrays.
2. Demonstrate how fundamental high-level programming constructs are implemented at the machine-language level.
3. Write assembly language program that can input, process and output results.
4. Demonstrate an understanding of basic computer architecture
5. Describe how synchronous digital systems are created using Digital Logic Gates.

Program Learning Outcomes:

- An ability to apply knowledge of computing and mathematics appropriate to the discipline;
- An ability to analyze a problem and identify the computing requirements appropriate for its solution;
- An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs, and use current techniques, skill, and tools necessary for computing practice;
- An ability to apply mathematical foundations, algorithmic principles, and computer science theory to the modeling and design of computer-based systems in a way that demonstrates comprehension of the trade-offs involved in design choices.

Prerequisites by Topic:

Class Prerequisite: CSE 30.

Topics:

- Number systems and math: binary, hex, octal
- C, I/O, structs, pointers, memory management
- Pointer arithmetic, arrays, C strings
- Debugging with GDB
- Compiling / Linking
- MIPS assembly language
- Machine Organization
- Memory Allocation and Management
- CPU Design.

Class/laboratory Schedule:

Lecture:	MW 1:30-2:45pm, SSB 170
Lab:	02L M 4:30-7:20pm, SCIENG 138
	03L T 7:30-10:20pm, SSM 154
	04L F 7:30-10:20am, SCIENG 100

Midterm/Final Exam Schedule:

This schedule is subject to change, but is tentatively set as follows:

- Midterm 1: During Week 8 (10-OCT - 16-OCT)
- Midterm 2: During Week 13 (14-NOV - 20-NOV)
- Final Exam: 11-DEC, S 11:30-2:30pm

Assessment/Grading Policy:

Participation	15%
Homework	20%
Lab Assignments	15%
Project Assignments	15%
Midterms 1, 2	20% (10% each)
Final Exam	15%

Coordinator:

Santosh Chandrasekhar

Contact Information:

My email: schandrasekhar@ucmerced.edu
I will try to answer your emails within 48 hours. However, I may not be able to answer emails after 5:00 p.m. or during weekends/holidays. Please plan accordingly.

TA's email:
Akshay Bhatia - abhatia8@ucmerced.edu
Hsin-Ping Huang - hhuang79@ucmerced.edu

Office Hours:

Santosh Chandrasekhar: T 9:00-11:00am, SE2 273 and via Zoom
Akshay Bhatia: W 11:00-12:00pm, SE2 Lobby and via Zoom
Hsin-Ping Huang: M 7:30-8:30pm and F 10:30-11:30am, SE2 Lobby and via Zoom

You may, of course, make an appointment to meet with the instructor or TA if necessary.

Course Policies:

CSE031 is a 4-credit course, which includes 2.5 hours of lecture, 3 hours of lab, and various assignments each week. You should plan on spending at least 6~8 hours outside of lecture and lab on reading and assignments.

Student Responsibilities: Please be sensitive to the learning environment. It is assumed that every student is attending lectures to learn; therefore, anything which distracts any student from learning is not appropriate behavior (for example, conversing during lecture, inappropriate comments, etc.).

In attempting to keep with a business-like, professional atmosphere, any behavior which would be considered inappropriate in a business setting will be addressed during lecture.

Use of Student Work: Assignments submitted by students may be used as examples for future students for educational or academic purposes. Names will be removed as possible. You may specifically request to not participate.

Labs: Each lab session has three components. The first component consists of peer-based collaborative discussion sessions (called Think-Pair-Share activities), which are moderated by the TA. The goal of the discussion is for you to work with your peers to collaboratively come up with solutions to various problems. The second component consists of TA's office hours to help with lab/project programming assignments. The third component consists of demo sessions where you will demonstrate your code to the TA. See the section on **Work Submission and Demonstration Policies** to understand what is involved with demonstrations.

There will be around 10 lab assignments this semester, where you will be implementing various programs in C and assembly language. Each lab assignment is **due** at 11:59pm on the 7th day after it is assigned. The assignment is **available until** 11:59pm on the 14th day after it is assigned. The period between the due date and available until date is called the **grace period**. The assignment will go offline at the end of the grace period, so students will no longer be able to submit (or demonstrate to receive credit).

There will also be 2 projects this semester. The due date will be longer compared to the lab assignments with no explicit grace period. However, depending on circumstances such as TA's availability, you may be allowed one additional week past the deadline for demonstrations only (code must still be submitted before the due date).

Collaboration Policies: All lab and project assignments should be done in compliance with the **Academic Integrity Policy** stated later in this syllabus. Copying even part of the code from the sources like StackOverflow, chegg.com, geeks-for-geeks, etc., is **prohibited**. We will enforce the academic integrity policy very strictly.

For LAB ASSIGNMENTS, you may work together with other students if you wish or when assignment asks for explicit collaboration. For PROJECTS, each student must write their program as an individual or in pairs. Please note that for the lab/project assignments we run your code through a system to detect similarity with other projects submitted by your classmates and a database of previous five (5) year's submissions and also code available in public repositories mentioned above. The algorithm analyses the structure and flow of the code, so simply changing the variable names and introducing minor changes to input/output will not be effective to defeat it. You would have to modify the code so much, that you are better off writing the code on your own. So, **caveat emptor**.

If your program is flagged as very similar to any program other than your group partner's program (you are allowed to work in a group of max. 2 students) and cannot explain why during your demo of your lab/project, you will face the consequences of plagiarism as listed in the **Academic Integrity Policy**.

When collaborating, you may talk with other students about general approaches to the problem, but you may not allow others to see your code, nor may you ask to see another student's code. To prevent your program from being flagged, follow the suggestions below when writing your program:

- Avoid writing programs as a group, unless you are working as one group or practicing paired programming. Even so, you need to implement your portion of the program alone.
- It is fine to collaborate with one another in terms of solving a problem. Once you have reached a solution with an abstraction of pseudocode, you should complete your pseudocode and implement the code alone. This will ensure that your code won't look similar to others' code.
- It is fine to use third-party source code as part of your program; however, the source code must be open-sourced and you must give proper credits to the sources (as comments). Keep in mind, your program is not your work if it is mainly composed of someone else's' work. Just like a paper you write that contains only quotes from others is not really considered as your creative work (**that's plagiarism**).

You may, of course, seek assistance from your TAs and instructor for all the assignments. There are also tutors available from PALS.

Work Submission and Demonstration Policies: Projects and lab assignments will be submitted through CatCourses. You can submit your work as many times as you want until the assignment is available. The last version of your submission will be evaluated for grading (with the corresponding penalty, if any, as stated in the DEADLINE AND LATE POLICIES below). To ensure that your submission is graded, you are REQUIRED to demonstrate your code to your TA or instructor BEFORE the available until date (the date when the assignment goes offline or the end of your grace period as defined earlier).

Submissions without a demonstration will receive a **grade of ZERO** for the individual coding portion of the assignment (you may receive partial credit for the answers to the Think-Pair-Share activities).

During demonstration, you will be asked to perform a walkthrough of your code that involves presenting your program in a step-by-step manner to the TA or instructor and answering any assignment-related questions that are posed to you. Questions about your lab assignments or projects can be wide-ranging. For instance, you can be asked to explain portions of your code in detail, provide reasons behind your decisions and choices, predict program behavior when modifications are introduced, etc. These questions will be used to test your knowledge of programming concepts relevant to the lab assignment or project being evaluated.

Deadline and Late Policies: The posted deadlines on CatCourses will be the official deadlines for each assignment. As mentioned earlier, you will be given a grace period (7 DAYS for labs) for late submissions past the due date. The instructor has the discretion to change this grace period on a per-assignment basis. If you submit an assignment past the due date but before the available until date (in other words, during the grace period), we will **take off 10% of the total value**.

No late work past the grace period will be accepted. Exceptions to this policy can only be made for valid reasons, with documentation. If you know before an assignment or project begins that you will not be able to make a deadline, please make arrangements with the instructor ahead of time. Similarly, make-up exams will NOT be provided unless arrangements are made beforehand.

Attendance: You are expected to attend the lab session for which you are enrolled, unless you make explicit arrangements with the instructor. Lab sections are where you will get most of the information and learn, so it is important to attend every week. Your participation grade will be a direct reflection of your lab attendance, in addition to textbook activities.

You are also expected to attend the lecture sessions.

Reasonableness: The "Reasonable Person Principle" applies throughout this course. This principle simply states that a reasonable request made in a reasonable way shall be reasonably handled by reasonable persons. The TA and I are reasonable people: we expect that everybody else involved in this class will be as well.

Disability Statement:

Accommodations for Students with Disabilities: The University of California Merced is committed to ensuring equal academic opportunities and inclusion for students with disabilities based on the principles of independent living, accessible universal design and diversity. I am available to discuss appropriate academic accommodations that may be required for student with disabilities. Requests for academic accommodations are to be made during the first three weeks of the semester, except for unusual circumstances. Students are encouraged to register with Disability Services Center to verify their eligibility for appropriate accommodations.

Academic Integrity Policy:

The campus Academic Honesty Policy states:

"Academic integrity is the foundation of an academic community. Academic integrity applies to research as well as undergraduate and graduate coursework. Academic misconduct includes, but is not limited to cheating, fabrication, plagiarism, altering graded examinations for additional credit, having another person take an examination for you, or facilitating academic dishonesty or as further specified in this policy or other campus regulations.

***Cheating** is the unauthorized use of information in any academic exercise, or another attempt to obtain credit for work or a more positive academic evaluation of work through deception or dishonesty. Cheating includes, but is not limited to: copying from others during an examination; sharing answers for a take-home examination without permission; using notes without permission during an examination; using notes stored on an electronic device without permission during an examination; using an electronic device to obtain information during an exam without permission; taking an examination for another student; asking or allowing another person to take an examination for you; tampering with an examination after it has been corrected, then returning it for more credit than deserved; submitting substantial portions of the same academic work for credit in more than one course without consulting the second instructor; preparing answers or writing notes in a blue book before an examination; falsifying laboratory, or other research, data or using another person's data without proper attribution; allowing others to do the research and writing of an assigned paper (for example, using a commercial term paper service or downloading a paper from the internet); and working with another person on a project that is specified as an individual project.*

***Plagiarism** refers to the use of another's ideas or words without proper attribution or credit. This includes, but is not limited to: copying from the writings or works of others into one's academic assignment without attribution, or submitting such work as if it*

were one's own; using the views, opinions, or insights of another without acknowledgment; or paraphrasing the ideas of another without proper attribution. Credit must be given: for every direct quotation; when work is paraphrased or summarized, in whole or in part (even if only brief passages), in your own words; and for information which is not common knowledge. The requirement to give credit applies to published sources, information obtained from electronic searches, and unpublished sources.
***Collusion** is when any student knowingly or intentionally helps another student to perform any of the above acts of cheating or plagiarism. Students who collude are subject to discipline for academic dishonesty. No distinction is made between those who cheat or plagiarize and those who willingly facilitate cheating or plagiarism."*

More information about the policy and the Office of Student Conduct can be found here:
http://studentconduct.ucmerced.edu/

Some students may still have some confusion (albeit the policy is quite clear), in particular concerning collaboration. The following rules are in place to make this issue clearer, from the perspective of my class.
Cheating vs. Collaboration: Collaboration is a very good thing. On the other hand, cheating is considered a very serious offense. **Please don't do it!** Concern about cheating creates an unpleasant environment for everyone. If you cheat, you risk losing your position as a student in the college. The school's policy on cheating is to report any cases to the university judicial office. What follows afterward is not fun. So how do you draw the line between collaboration and cheating? Here is a reasonable set of ground rules. Failure to understand and follow these rules will constitute cheating and will be dealt with as per university guidelines.
The Simpson's Rule: This rule says that you are free to meet with a fellow student(s) and discuss assignments with them. Writing on a board or shared piece of paper is acceptable during the meeting; however, you should not take any written (electronic or otherwise) record away from the meeting. This applies when the assignment is supposed to be an individual effort or whenever two teams discuss common problems they are each encountering (inter-group collaboration). After the meeting, engage in a half-hour of mind-numbing activity (like watching an episode of the Simpsons), before starting to work on the assignment. This will assure that you can reconstruct what you learned from the meeting, by yourself, using your brain.
The Freedom of Information Rule: To assure that all collaboration is on the level, you must always write the name(s) of your collaborators on your assignment in the beginning of your submission file as a comment.

Computer Science Department Policy on Academic Honesty

As stated in the campus-wide Academic Honesty Policy (AHP), "academic integrity is the foundation of an academic community". Accordingly, the CSE faculty takes this matter very seriously and has embraced a zero tolerance on this matter. The process described in the following establishes the minimum consequences for violations of the AHP in CSE courses, but repercussions may be more severe for egregious violations. The Computer Science Department Policy on Academic Honesty ("CSE Policy" from now onwards), does not substitute the AHP but rather specifies how it will be implemented when students enrolled in classes offered by the Computer Science and Engineering (CSE) department are found in violation of the AHP. In particular, the CSE Policy defines how the CSE faculty implements the "Instructor-Led Process" described in AHP 802.00.A. This policy and the associated processes have been developed in collaboration with the Office of Student Conduct and the School of Engineering and is jointly implemented by the CSE Faculty, the School of Engineering, and the Office of Student Conduct. The CSE Policy has been in effect since the Fall 2019 term.

Preamble: Computer science education relies on a variety of methods to assess students' preparation and learning. The term "assignment" shall be interpreted as any method or process resulting in a grade or contributing to the final grade for a class. Accordingly, the term "assignment" used in the following includes, but is not limited to: homeworks, quizzes, in-class exams, take-home exams, programming assignments, software projects, and presentations.

Shared Responsibility: Maintaining an environment where academic integrity is valued and enforced requires commitments by both instructors and students. Instructors will specify what type of collaboration is allowed or

disallowed for a given assignment, and students should strictly follow the provided guidelines. When in doubt, students should contact the instructor and ask for clarifications.

First Infraction: If it is determined that a student has cheated, plagiarized, or otherwise violated the AHP, the student will receive a 0 (or equivalent grade) for the assignment. As per the AHP, violations will be reported to the Dean of the School of Engineering and the Office of Student Conduct for review of possible violations of the Code of Student Conduct.

Additional Infractions: The School of Engineering keeps a record of all infractions reported by its faculty. If upon receiving a notification it is determined that the student has one or more prior violations of the AHP, the School will inform the instructor who reported the new violation. The additional violation will immediately lead to a failing grade (F) for the course. The student will be informed in writing and will not be allowed to withdraw from the class. According to CSE Policy, students should note that even the first infraction in a class may lead to a failing grade if after reporting it is determined that the student had been previously sanctioned for one or more infractions in other classes. Students will have the right to appeal the instructor's decision as per AHP 802.00.A.

Resources: If in doubt, students are encouraged to seek guidance from the faculty, advisors, and the Office of Student Conduct. Additional resources can be found on:

- http://studentconduct.ucmerced.edu/
- https://ombuds.ucmerced.edu/
- https://eecs.ucmerced.edu/sites/eecs.ucmerced.edu/files/page/documents/computer_science_department_policy_on_academic_honesty_fall_2019.pdf