COMP 122/L: Computer Arch. and Assembly Language Summer 2022

Instructor: Kyle Dewey (kyle.dewey@csun.edu)
Course Web Page: https://kyledewey.github.io/comp122-summer22/
Office: JD 4419 (I will not physically be there); Zoom link for office hours posted on Canvas and available via email.

Special COVID-19 Message

The course is being run as a synchronous online course. We will never meet in person, though we will meet virtually via Zoom regularly each week at the assigned time (see SOLAR for meeting time details). Assuming you're enrolled in the course or on the waitlist, I have emailed the Zoom link to you. If somehow you do not have the Zoom link, email me at kyle.dewey@csun.edu. For exams, if you cannot be on Zoom during the scheduled exam time, let me know ahead of time for alternative arrangements.

The synchronous lectures will be recorded and made available through Canvas, in case students want to view them asynchronously. These lectures will only be accessible to CSUN students either enrolled or waitlisted for the course. However, **by enrolling in, or waitlisting this course, you consent to having any voice or webcam recorded.** That said, **I will never require you to use your webcam or speak**; only what you voluntarily send will be recorded. Questions can either be asked verbally or textually through the Zoom chat. I will verbally repeat any questions in the chat before answering them, but I will not identify who said the question. Even so, it's possible that your name will end up in the meeting recording, identifying you as a participant (Zoom sometimes unavoidably shows participant names in the recording).

Course Description (from the catalog)

Introduction to computer architecture, assembly language programming system software and computer applications. Number systems and data representation. Internal organization of a computer. Primitive instructions and operations. Assembly language.

Learning Objectives

A successful student will learn basic assembly programming skills, understand the fundamentals of digital logic design, and understand the interface between the two. The MIPS instruction set will be used for assembly, though the concepts are broadly applicable. In particular, successful students will be able to:

- Describe how modern computers represent numbers, and interconvert between different numeric representations
- Perform common operations over computer-represented numbers, and design circuits which perform these operations
- Write programs in MIPS assembly, including conditionals, loops, arrays, and functions
- Design and simplify combinatorial circuits with Boolean algebra and Karnaugh maps (K-maps)

- Design sequential circuits, including those implementing finite state machines
- Understand, from a high-level, the main components of a high-level processor
- ...among many others

Course Motivation and Goal

A program doesn't have much utility until it runs on some piece of hardware. Here, we look at the low-level details behind execution, specifically the interface between hardware and software. Knowledge of these details can help you understand performance issues in your code, as well as writing low-level code like embedded software or operating systems. Ultimately, the goal of this class is to give you a basic working knowledge of these low-level details.

Textbook

No textbook is required. If you'd like a textbook for further study, two decent supplemental textbooks are:

- Computer Organization and Design: The Hardware/Software Interface (David A. Patterson and John L. Hennessy); any edition from the past several years
- Computer Systems Organization and Architecture (John D. Carpinelli)

Grading

You will receive a **single combined grade** for the lecture and lab. Your grade is based on the following components:

Lab Assignments	30%
Midterm #1	20%
Midterm #2	20%
Final Exam	30%

There will be roughly two lab assignments each week. The exact number of lab assignments has not been set, as this will depend somewhat on how the class progresses. Lab assignments are submitted through Canvas (https://canvas.csun.edu/). In the event that there is a problem with Canvas, you may email your assignment to me (kyle.dewey@csun.edu), though this should be considered a last resort.

Plus/minus grading is used, according to the scale below. The left column shows the minimal score necessary to receive the grade in the right column. The highest letter grade possible given the score is chosen; e.g., if you receive an 88.2, you'd receive a 'B+' for the course, which corresponds to being >= 86.5.

If your score is >=...	...you will receive...
96.5	A+
92.5	A

If your score is >=...	...you will receive...
89.5	A-
86.5	B+
82.5	B
79.5	B-
76.5	C+
72.5	C
69.5	C-
66.5	D+
62.5	D
59.5	D-
0	F

Collaboration for Assignments
All students are required to submit their own individual work. For assignments (and **only** assignments), students may discuss among each other, as long as they don’t digitally share code. That is, you **cannot** simply email your code to someone else. However, you **may** discuss your actual code with someone else, including merely viewing code. The only stipulation is that **if you do discuss with someone else, say so in your submission.** This is not for punitive reasons; this is only so I get a sense of who is working with who. My intention with this policy is to enable collaborative learning, as opposed to simply sharing a solution.

Plagiarism and Academic Honesty
While collaboration is allowed on assignments, you are responsible for all of your own work. You may **not** take code from online sources and submit it as your own. If you must take code from online, cite where you took the code from. Worst-case scenario, you'll receive a 0 for whatever you took, but no further action will be taken. In general, code taken online which solves more general things (e.g., "how do I iterate through an array in Java") is more acceptable than code which solves more specific things (e.g., "how do I implement a recursive find function over immutable linked lists in Swift"). General bits of code only give you pieces of a solution, whereas specific bits of code often will give you a complete copy/pastable solution. If it's not 100% clear if something is permitted to be used or not, you can always ask me beforehand.

Chegg is specifically disallowed as an online resource, as it's almost always used as a repository of complete questions with answers. That is, the questions/answers are practically always of the specific kind mentioned above.

No discussion whatsoever is allowed during exams, except with the instructor. Any violations can result in a failing grade for the assignment/exam, or potentially failing the course for egregious cases. A report will also be made to the Dean of Academic Affairs. Students who repeatedly violate this policy across multiple courses may be suspended or even expelled.

Attendance

In the first week of class, I will take attendance. If you miss both sessions in the first week and have not made alternative arrangements with me, you must drop the class, as per University policy (http://catalog.csun.edu/policies/attendance-class-attendance/). After the first week I will not take attendance, and attendance is not mandatory, though you are strongly encouraged to attend. I enforce this policy in order to help students on the waitlist get into the course.

Communication

In general, any questions should be made through Canvas. You can also email me, though I'm usually much faster to respond to Canvas than my general email.

Late Policy / Exam Scheduling

Late assignments will be accepted without penalty if prior arrangements have been made or there is some sort of legitimate emergency (at my discretion). If you must be absent from an exam, contact me ASAP to see if alternative accommodations can be made.

If an assignment is otherwise submitted late, it will be penalized according to the following scale:

If your assignment is late by <= this many days...	...it will be deducted by...
1	10%
2	30%
3	60%
4+	100%

To be clear, assignments which are submitted four or more days beyond the deadline will not receive credit. The reason for such a harsh late policy is that we will generally discuss solutions in class shortly after the deadline, and this late policy discourages people from simply pulling a solution from an in-class discussion.

Class Feedback
I am open to any questions / comments / concerns / complaints you have about the class. If there is something relevant you want covered, I can push to make this happen. I operate off of your feedback, and no feedback tells me “everything is ok”.

Class Schedule (Subject to Change):

Week	Day	Content
1	Wednesday, 7/13	Syllabus, number representation
	Thursday, 7/14	Binary operations
2	Monday, 7/18	Binary operations
	Tuesday, 7/19	Floating point representation
	Wednesday, 7/20	Floating point representation
	Thursday, 7/21	General assembly and MIPS introduction
3	Monday, 7/25	MIPS arithmetic
	Tuesday, 7/26	MIPS branches / conditionals / loops
	Wednesday, 7/27	MIPS branches / conditionals / loops
	Thursday, 7/28	**Midterm #1 Review (exam online asynchronous)**
4	Monday, 8/1	MIPS memory / arrays
	Tuesday, 8/2	MIPS memory / arrays
	Wednesday, 8/3	Call stack / calling convention / MIPS functions
	Thursday, 8/4	Call stack / calling convention / MIPS functions
5	Monday, 8/8	Boolean logic and combinatorial circuits
	Tuesday, 8/9	Boolean logic and combinatorial circuits / simplifying circuits with Boolean algebra and K-maps
	Wednesday, 8/10	Simplifying circuits with Boolean algebra and K-maps
	Thursday, 8/11	**Midterm #2 Review (exam online asynchonrous)**
6	Monday, 8/15	Sequential circuits
	Tuesday, 8/16	Sequential circuits
	Wednesday, 8/17	Finite state machines

Week	Day	Content
	Thursday, 8/18	Finite state machines
7	**Monday, 8/22**	Finite state machines
	Tuesday, 8/23	**Final exam review (exam online asynchronous)**