SYLLABUS

COMP 122 Intro to Computer Architecture & Programming in Assembly Fall 2020

Instructor: Dr. Jeff Drobman

Dr Jeff's Web Page: http://www.drjeffsoftware.com
Class Web Page: http://www.drjeffsoftware.com/classroom.html
Email: jeffrey.drobman@csun.edu
Office Hours: In Lab, after class

Pre-requisites

C or better in: COMP 110 and in one of Math 103, 104, 105, or 150A

Course Description

Introduction to Computer Architecture via CPU organization, instruction sets (ISA), op codes, addressing modes, register and memory models, virtual memory, L1/L2/L3 caches, simple pipelining, memory-mapped I/O with interfacing using interrupts, and clocking. We shall emphasize equally hardware and software and focus on how they interface and interact.

The only way to explore Computer Architecture fully is by programming at the Assembly language level. We shall explore several CPU and MCU examples, but focus mainly on the MIPS (MIPS I and 32) vs. ARM (v5/7) ISA's, and compare to x86. We shall use an appropriate assembler/simulator IDE for each ISA (MIPS, ARM). Emphasis will be on mastering the unique constructs of assemblers, such as "macros", "defines" and memory segmentations (e.g., ".data", ".text"), along with learning the power of simulations and their tools (*MARS, ARMsim*).

Programming languages: Students are expected to already know how to program in at least one high-level language: Java (from COMP110), C/C++ or Python.

Other topics: Number systems and data representation; CISC vs. RISC; microprocessor (CPU) vs. microcontroller (MCU); bit-slice microprocessors (Am2900) and microprogramming; RISC instruction pipelines; ISA simulators; GPIO; PSW, state machines and clocking; basic logic design of registers and ALU's plus DeMorgan's Law for signal polarity. (Advanced topics outside the scope of this class will be included in COMP222: CPU pipeline details, superscalar, multi-core, multi-threading, speculative execution, branch prediction, security flaws like *Meltdown* and *Spectre*.)

Textbooks

There is no required textbook for this course, but there is a recommended one: an interactive e-book from zyBooks by Patterson and Hennessy, titled, Computer Organization and Design (MIPS ed.). This book will be required for COMP222, so it is advised to get familiar with the material now. The cost is $72 (but lower for a repeat use). There will also be an ARM textbook available as a free online resource.

Slides

PowerPoint slides are available for download from the Class Web Page (linked above) as PDF files, so need to buy or install PowerPoint itself. Additionally, on occasion, a few slides will be emailed to the class.

Software Tools

We shall use the "MARS" IDE and simulator for MIPS, and "ARMsim" for ARM programming and simulation. (Note: *MARS* is available for Win PC and Mac, while *ARMsim* is only available for Win PC. Mac users will be accommodated.)

Lab Hardware (Optional)

We will take a cursory look at the ***Raspberry Pi 4*** as an inexpensive yet functionally complete and powerful computer board, which comes with an advanced ARM CPU. The *Raspberry Pi 4* ("Pi") contains: ARM Cortex-A72 CPU (4-core, 1.5GHz); 1GB DRAM; 4 USB ports (2 and 3); 2 console ports (mini-HDMI); USB-C power connector (AC to DC adapter needed); Ethernet (10/100 on cat-5 connector, using a Broadcom PHY chip); 40-pin GPIO connector; connector for external camera + video/audio. There is a slot for an external SD card by SanDisk, with pre-installed Unix OS and compilers for Java and Python. Students are encouraged to buy their own, but it is not required. A complete system includes the Pi 4 computer board (about $35); plus AC to DC adapter, keyboard and mouse, and SD card (another $45).

Computers

We would normally use the Lab machines (Windows 10) as workstations, but we cannot while remote. Students must use their own PC's or Macs by installing MARS and ARMsim (not yet available for Macs). A few PC laptops are available for loan from our CECS IT group.

Computer Accounts (not necessary while remote learning)

Every student registered for the course has a networked account that can be used on all CS Department computers. Your account (user id and password) is the same as the one issued by the University. The instructor does not have sysadmin authority over student accounts. For problems logging in, see the CECS Information Services office in Room JD 1112, extension 3919. Note that most students will opt to use their own laptops, and will have no need to use the school's computers, but will still have access to the CECS network via WiFi.

Exams

There will be a Quiz (week 5), Midterm (week 8) and a Final Exam. These exams will be multiple choice entered on Scantron forms, while the final exam will add a Part 2 for writing code segments.

Assignments

This course is about studying architectures and ISA's via assembly level programming. There will be ~7 graded "Lab" assignments, plus 2 "Projects" (midterm and final). Project 1 is a research project, while Project 2 is a programming assignment using a simulated LED display. The Labs will be carefully selected to integrate the current lecture material and lead the student to a mastery of traditional and fundamental programming techniques, concepts and functions.

Lab Form

Submissions of Labs will use a provided "Lab Form" (Word doc) for documentation, including a source code listing. All assignments are to be submitted as files (Word .doc or .pdf) attached to emails sent to this instructor directly (using my csun.edu address above).

Late Submission Policy

All graded assignments will have a due date. You will not have a grace period. Late assignments will not be accepted without permission (for valid reason). No assignments will be accepted after the last week of class.

Lecture Material

All lectures will be delivered via PowerPoint slides (as PDF files), combined with live programming on the chosen IDE's. All slide sets will be updated and posted in PDF format to the class web page, with occasional new slides also sent via email. However, it is recommended that for the best level of preparation, you should attend lectures and participate in discussions, rather than simply reading the lecture slides on your own. (There would normally also be a limited amount of hand-drawn notes on the board, for which each student would be responsible for taking notes.)

Lab Work

This course is focused on the "Lab". The lab sessions will mostly involve the instructor going over the assignments, and running some example code live. Learning to use the IDE tools is an important part of the lab and the course. During the lab sessions, students will work on their assignments, with help from the instructor, or any other students. It will usually be necessary to continue to work on your graded programming assignments on your own time outside of your official lab time.

Save Your Work (when on campus)

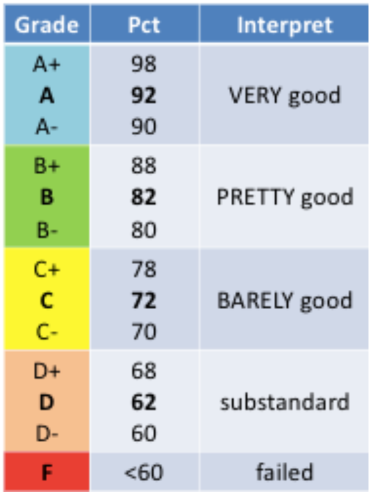
Work done in the lab on the class workstations can be saved to your personal directory on the CECS file server (the "Z drive"). Do not store anything of importance on the hard drive of the local machine, since the machines can be reformatted at any time without prior notification. In addition to Z drive storage, you can use a USB flash drive to store backup copies of work performed in the lab, or you can send an email to yourself with programming source code as an attachment.

Tutors

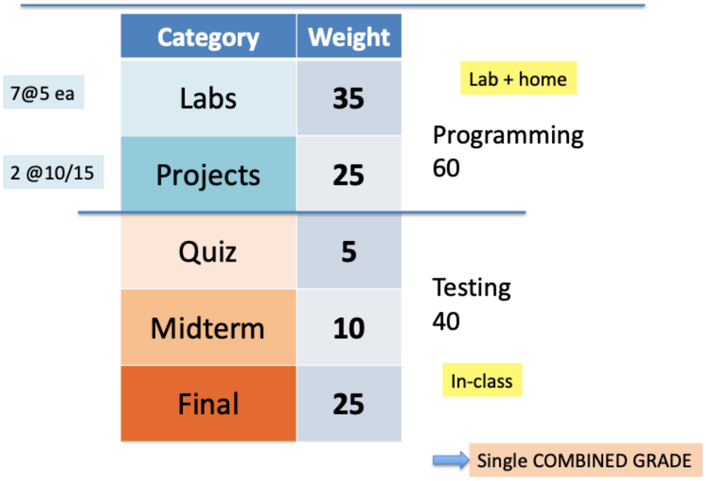
The School (CECS) normally provides a group of tutors available M-F in the designated tutoring room in JD1622. Students will be expected to make use of this resource. (This is subject to change for remote learning.)

Grading

You will receive **Plus/Minus** grading, as shown (except there cannot be an A+). The exams and Labs will be weighted as shown.



Grade	Pct	Interpret
A+	98	VERY good
A	**92**	
A-	90	
B+	88	PRETTY good
B	**82**	
B-	80	
C+	78	BARELY good
C	**72**	
C-	70	
D+	68	substandard
D	**62**	
D-	60	
F	<60	failed



	Category	Weight	
7@5 ea	Labs	**35**	Lab + home
2 @10/15	Projects	**25**	Programming 60
	Quiz	**5**	
	Midterm	**10**	Testing 40
	Final	**25**	In-class

Single COMBINED GRADE

Attendance

For at least the first several weeks, attendance will be closely monitored. Frequent unexcused absences may result in a grade penalty. After the first few weeks of the semester, attendance will not be monitored as closely, but note that the scheduled lab time is the most appropriate time to ask questions and get help on your assignments. I am not willing to spend an unreasonable amount of time outside of class (including excessive use of email) to explain class material to anyone who is not attending the lab. In other words, you are expected to participate in lab discussions by attending class, and get programming help on your assignments in person by attending the labs. Students are welcome to bring personal laptops with a wireless connection to the lab.

Personal Behavior in the Classroom

You are expected to arrive to the class on time, and to wait until dismissed before leaving. You are expected to be polite, pay attention, and participate in discussions: no web surfing, no private conversations, no cell phone interruptions, no sleeping/snoring, etc. You may quietly excuse yourself for bathroom breaks, to take important phone calls, and to address other urgent personal business, without asking for permission, but keep interruptions to a minimum. During labs, you are expected to keep personal web surfing to a minimum (quickly check your email then get to work). You may not play music or otherwise disturb other students while in the lab. If you finish your lab work early, you may remain in the lab and work quietly. You may be asked to leave after you finish your work if your personal computer usage is disruptive to other students. If you're ahead of the rest of the class and finish your work early, kindly consider looking around and finding someone who could use your help. You may not invite friends who are not enrolled in the course to join you in the lab to "just hang out".

Plagiarism and Academic Honesty

Academic honesty is expected from all students taking the course. All work that a student submits for course credit must be performed by the student who submits it. Use of for-pay programming services such as RentACoder for programming projects for course credit, either as a buyer or a seller, is not permitted, and is considered a serious offense. Exams must be taken according to parameters established by the instructor at the start of the exam. In general terms, students are prohibited from all communication, both verbal and electronic, with anyone except the instructor during an exam. Notes and other reference aids may or may not be permitted, instructor will specify what resources are allowed, if any, when exam is announced. For programming projects, you may discuss coding details with other students, but you cannot copy and paste large chunks of code from one student's work and then submit it as your own. Violations of this policy can result in a failing grade for the project or exam in question, and depending on the severity of the violation, may result in a failing grade for the course, as well as the violation being reported to the Dean of Academic Affairs. Students who repeatedly violate the policy across multiple courses may be suspended and even expelled from the University.