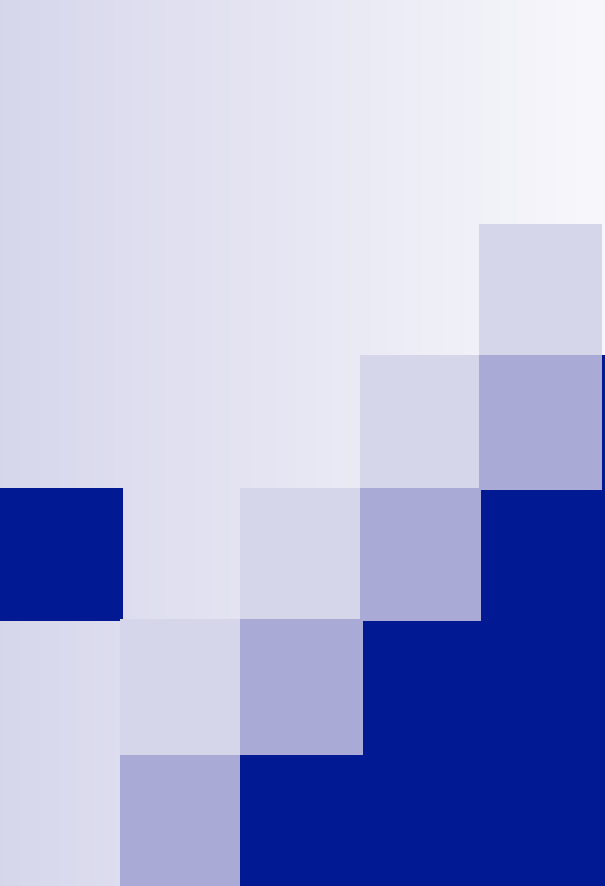




Introduction

ICS312 Machine-Level and Systems Programming

Henri Casanova (henric@hawaii.edu)

Course Goal



- At this point in your computer science education, most of you have only a very high-level understanding of how a computer run programs
- Your experience so far: you write/generate code in some language, you build it (if a compiled language), and then you run it
 - Each of these steps involve hitting buttons in IDEs or typing some Shell commands
- Most of you have only very superficial understanding beyond “this magically creates an executable”, “this magically runs an executable, which my CPU magically knows how to do”
- A big part of being an effective (and employable) computer scientist is knowing what the magic is, which is the goal of this course



What we will learn/do

- Learn what's inside a computer and CPU
 - Very basic elements of computer architecture
- Gain exposure to assembly programming
- Use assembly as a vehicle to understand/internalize fundamental concepts that apply to high-level programming/computing
- Learn the concepts and techniques behind compilers
 - And other “tools”, like linkers and debuggers



Course Website

- Located at:

- http://courses.ics.hawaii.edu/ics312_fall2026/
- Linked from my personal homepage
 - Google for “Henri Casanova”

- Organized as Modules

- All lecture notes as PDF files
- Sample/Practice assignments
- Pointers to useful on-line material
- Announcements
- A link to the Syllabus
 - Which we’re going over now in these slides

- Let’s look at the Web site...

Textbook(s)

- The main text is a **free** book:
 - *PC Assembly Language*, Paul A. Carter
 - Available for download on the course's Web site
- The user's manual for our assembler, NASM, is also available on the course's Web site, and other manuals will be made available throughout the semester
- Other interesting free resource available for download:
 - *The art of assembly programming*, John W. Lockwood
 - *Assembly Language for x86 Processors*, Kip Irvine
- See links from the Syllabus on the Course Web site



Lectures and Office Hours

- Lectures in CR 115, Mon/Wed
10:30AM-11:45AM
- Lecture notes are posted on the course's Web site regularly
 - You can read them before or after the lecture, up to you really
- Office hours, TA, etc.
 - All information on the Web site



Discord

- It is **paramount** that you register on the Discord server for this course
 - E-mail was sent out
- Discord is used for announcements/news/polls

Inverted Lectures

- A few lectures will be “inverted”
 - You watch a screencast at your own pace
 - The lecture period is for questions and practice exercises
- I do this for a few topics in the course that are more “mechanical” or “difficult”
- You must watch the screencast ahead of time!
 - E-mails reminders will be sent out



Screencast Lectures

- A few lectures will be **screencast**
- This semester, I am required to travel to Washington DC during the week of Sept 14, so the **Mon 9/14 and Wed 9/16 lectures** will be screencast

Web site Content

- In spite of my best efforts it happens that the course Web site could have small problems (typos, missing link, etc.)
- Anytime you see anything strange/broken on the Web site, or missing lecture notes, please let me know right away!
 - A one-line e-mail, a DM on Discord, etc.



Grading on 1000 points

- Sample homework assignments for 0 points
 - Solutions are provided
- Four exams
 - Midterm #1 (200 points)
 - Midterm #2 (200 points)
 - Midterm #3 (200 points)
 - Final exam (330 points)
- Each exam assumes knowledge of the material up to that point in the semester (i.e., “cumulative”)
- Quizzes (70 points)
 - 8 10-point quizzes, worst grade is discarded

Exams

- Exams are taken in class, closed-note
- Pocket calculators, not programmable calculators or phones, allowed (but not needed)
- Exams are randomly generated and students have different exams
 - So don't cheat with your neighbors, it's super obvious if you do (and it still sometimes happens!)
- Each exam will have mostly exercises that match exactly in-class, practice, and/or sample homework problems
- The final exam will include, among others, one exercise about material from each of the previous midterm exams

Quizzes

- 8 Quizzes in the semester
- Taken on the first lecture day of the week
 - Always on a Monday, unless that Monday is a holiday, in which case it will be on a Wednesday
- Always **announced the previous week ON THE COURSE'S WEB SITE**
- Taken at the **beginning of the lecture period, in the first 10 minutes**
 - You **cannot take the quiz if you show up more than 5 minutes late** to the lecture
- No make-up quizzes, unless a documented reason
- But **the worst quiz grade is discarded**

Learning during AI chaos

- Common question: “Now that anybody can vibe-code anything, what’s the value added by a CS bachelor degree?”
- A CS degree was always supposed to be about so much more than just coding (algorithms, data structures, systems, how things work under the hood, etc.)
- But for decades it has been possible for a CS graduate’s only value to be that they could hack some code
 - Because the tech industry has had such a high demand for coding for so many years
- AI has changed all that
- So a CS degree should (once again!) be about fundamental concepts, how they fit together, how things work
 - Many companies are going back to pencil-paper / knowledge interviews
 - Interviewers will immediately understand what you know by watching your prompts when LLMs are allowed during the interview
- **Bottom line:** The only winning move now is to be more than a vibe coder: actually understand fundamental concepts and how things work!
 - And then use AI as a person who knows things and can be discerning!
 - Have your prompts be as good as “old people prompts” :)

ICS312 in the era of LLMs

- Graded programming assignments are no longer possible
- Ungraded programming assignments (with solutions) are provided
 - Experience shows very few students turn in optional assignments anyway
- So without students being forced to write code, the amount of “let’s learn how to write assembly” content in this course is reduced
 - We ONLY learn enough assembly to understand key concepts
- The big question: “Can students internalize concepts without going through the work of doing homework assignments?”
 - For some of you, yes. For some of you, you will have to force yourselves to do more practice
 - The assumption in this “new world” is that students are adults and know what they need to do to pass exams
 - A lot more in “on you” in this new era 😬
- Fear not: we do quite a bit of practice and live-coding in class!
 - You should never hesitate to ask “can we live-code this?” during lecture

In-class Participation, Shyness (1)

- A big problem in this course recently has been lack of in-class participation
 - Yet, I keep doing in-class exercises/questions, live-coding, and I think I encourage questions as much as possible
- It's a problem for many reasons
- Reason #1: It makes lectures more dull
- Reason #2: It's hard for me to tell when to repeat an explanation, or slow down
 - Several CES evaluation comments last semester said: "sometimes he goes too fast, and I get lost, but I am too shy to raise my hand"
 - I just cannot say "should I explain this again" after each explanation I give!
- Reason #3: I remember about 5 students from last semester (the few who spoke up), and I serve as a reference or write a letter of recommendations all the time
 - A "took my class, got an A, scored in the top 10% of students in the class, don't know who they are" is typically a terrible recommendation letter for grad school (especially if you expect funding, etc.)
 - I get call from people for references, and I says "that person is a row in a spreadsheet, a pretty good row, but I have nothing to tell you"

Sample Post-Semester Interaction

Thank you, Professor. I completely understand. I'm a relatively shy and quiet student, so I didn't interact or participate as much as I probably should have.

I really appreciate that you're still willing to write a recommendation letter for me.

Of course, be aware that my letter will basically only say "did great in my course". A better letter writer would be somebody with whom you did a project, and in whose class you had a very high level of in-class participation. Do you still want me to write you a letter?

Henri

On Jul 8, 2026, at 4:27 PM, REDACTED wrote:

Dear Professor,

I was a student in your ICS 312 course and earned an A in the class. I am currently preparing my application for a Master's program in Computer Science and was wondering if you would be willing to write a letter of recommendation for me.

In-class Participation, Shyness (2)

- Reason #4: Your professional lives will entail speaking up or lose out
 - Personal rant: I am tired/depressed with “shy” people being passed up on for opportunities all the time
 - <story>Tell students Edward’s story about the engineering competition in California!</story>
 - And, sadly, as we know, opportunities are not plentiful these days
- Hopefully, this semester will be different?
 - I am cautiously optimistic because ICS332 wasn’t so terrible last semester.... but ICS312 was two semesters ago...
 - I am open to suggestions of course
 - Part of the problem of “nobody answers any question”, is students not paying attention during lectures, I get that, but that’s another problem altogether

CES Evaluation

- Extra credit given to all students:
 - 0 points if CES completion rate is $< 80\%$
 - 5 points if CES completion rate is $\geq 80\%$
 - 10 points if CES completion rate is 90%
- Why?
 - I do look at the student comments every semester and evolve the course accordingly
 - Even if you love the course, it's important for me to hear what things didn't work
 - These evaluations have more impact than you may think and are taken seriously
 - Impact for individual faculty, for the whole department, for future students, etc.
- Last semester: $< 50\%$ completion rate!!

How to not do well in this course?

- **Last semester:** only 70% of students who started the course ended up passing it
- **Don't come to class**
 - Some students I will never see in class, and that's "fine", but: ~50% of those students fail, 25% ace everything, and 25% barely get a C
- **Don't look at the sample homework assignments**
- **Don't practice the "practice problems" before exams (???)**
- **Cheat**
 - Cheating is bad for many reasons, including hurting the reputation of ICS graduates!
 - **If you are caught cheating or enabling cheating:**
 - zero on the exam
 - overall grade lowered by a step (i.e., a "B" becomes a "C")
 - reported to UH's Office of Judicial Affairs (as required)
- **Expect that "what can I do for extra credit to pass this course?" at the end of the semester will be met with a positive response**
- **Don't study for the quizzes**
 - "It's only a small fraction of the grade"
 - But studying for quizzes is a HUGE help to prepare for exams
 - When I don't do quizzes, exam scores drops!

Software/Hardware for ICS312

- We'll have to use an Intel-based machine that speaks the x86 32-bit Instruction Set Architecture (IA-32)
 - Luckily, we all have that (or can emulate that)
 - In fact we have 64-bit machines!
 - But in this course I use 32-bit assembly because:
 - It has less “stuff” and is better for a first experience
 - We learn exactly the same set of things
 - Our free textbook is about 32-bit assembly
 - If you need to go to 64-bit assembly, the transition is easy
- We'll use the NASM software package, which is also free
- Let's look at **Homework Assignment #0**, which is ungraded but necessary if you want to do any coding in assembly



More Questions

- Any questions on the syllabus?
- Any questions on the course in general?



Participation Verification

- As you know, each instructor has to report on “Student Participation” and certify the class roster
 - If you have not “participated”, you could be dropped from the course
- **IMPORTANT:** Do the ungraded “Participation Verification” Assignment **posted on Lamakū** by **THIS FRIDAY, 11:59PM**

What's Next?

- There is already an announcement on the course Web site regarding **two short screencasts to watch BEFORE THE NEXT LECTURE**
- These screencasts are about “numbers and computers”
 - Many of you will find them easy and can fast-forward a lot of the material (or just look at the lecture notes quickly)
 - Make sure you watch them before this coming Wednesday as we'll do in-class practice exercises to make sure we're all up to speed
- This is all in the Integers and Computers module...
 - Let's look at it now...
- Do **Homework #0** if you want to write any code this semester
- **Let's finish today with an ungraded quiz right now!**