

EECS 370: Intro to Computer Organization

Fall 2024

Course Overview

This course is intended to give you a basic understanding of how computers execute programs. Understanding computers means understanding the hardware/software process of how you and the computer work together to have the computer carry out a task.

In your introductory programming courses (e.g. EECS 280), you learned how to express a task in terms of a high-level programming language like C++. In EECS 370, you will see how a low-level language is executed by the hardware, and you will see how to put together basic hardware building blocks to form the functional units of a computer.

To achieve these goals, you will design and "build" simple computers at various levels of detail. In this course, building will not mean connecting chips and gates. Rather, you will describe the hardware in diagrams, finite-state machines, and hardware simulators (written in C).

Learning Objectives

- Understanding the basics of how computers execute programs
- Know the hardware building blocks which form the functional units of a computer
- Ability to describe computer hardware through diagrams, finite-state machines, and hardware simulators (expressed in programs written in the C programming language)
- Ability to create programs in a low-level language (expressed in Assembly language)

Prerequisites

Students must have taken EECS 280. From that course they should have learned and be comfortable with programming C++ on Unix. A brief overview of C vs. C++ will occur during the first lab section of the semester. Also you can go to staff office hours for additional help.

In addition, students should have taken either EECS 203 or EECS 270.

Quick Links

Administrative Requests	Turning in Projects	Exams
Lecture Format	Project Late Days	Textbook
Lab Format	Project Grading	Computers
Grading Policy	Academic Integrity	Course Etiquette
Class Projects	Homework	Right to Revise

Administrative Requests

Please let us know as soon as possible with an Admin Request if you experience interruptions to your studies and will need accommodations (extensions, extra drops, etc.) for homeworks, labs, projects, or exams. **Students do not need to submit requests for lecture absences, as attendance is not required or graded.** All request forms (and respective deadlines) are listed on the course website (eecs370.github.io).

For **urgent issues** (needing a response in <12 hours), submit an Admin request AND email eecs370admin@umich.edu and please include "EECS 370" in the subject line

Lecture Format

There are 3 sections that are traditional 80-min live, in-person lectures. Students enrolled for section 003 or 005 are encouraged to attend one of the live in-person lectures. All lecture sessions will be recorded and released on the website. There is also a set of asynchronous studio recorded lectures linked on the website.

Lecture covers material which will help prepare you for the Homework Assignments, Projects, and Exams. Attendance, while not required, is critical for success in the course.

Lab Format

Lab sections will be in-person and involve group assignments that are graded for correctness or completion. Staff will form groups after the first week of class based on student survey results. **Attendance is required at your registered section in order to receive credit for the assignment** and counts for 40% of the lab grade. Slides and assignments will be released beforehand.

When calculating your course grades, we will grade the lab assignments out of **80%** (capped at full credit). So, for example, if you scored 33/50 on lab 1, your effective score is $(33/50)/80\% = \sim 83\%$. If you scored 40/50 or higher, your effective score is 100%. This grading policy is intended to take off some of the time pressure and overhead of working in a group within the lab setting. A few minor mistakes that slip through the cracks will not impact your final score if you get everything else correct.

As an incentive to keep up with lecture material, 20% of the lab points will be determined by completing a small quiz online covering the relevant lecture material. This must be completed ~~Wednesday~~ **Thursday** evening before the lab by 11:55 pm.

Total lab score = $(40\% * \text{attendance}) + (40\% * \text{adjusted assignment score}) + (20\% * \text{online quiz})$

Grading Policy

Final grades will be based on the total points earned on homework, projects and exams. The grade distribution is as follows:

Category	#	%	Notes
Projects	4	40% (10% each)	Projects split into multiple parts
Individual Homework	4	5% (1.7% each)	Drop lowest score
Labs	12	5% (.5% each)	Drop lowest two
Midterm	1	25%	
Final exam	1	25%	

After computing the total weighted score, we use these ranges to assign letter grades. Each range is inclusive at the bottom and exclusive at the top; for example a score of 88.9% is a B+ and a score of 90% is an A-. We do not round scores.

Total Weighted Score	Letter Grade
00 - 56.99...%	E
N/A	D-
57 - 59.99...%	D
60 - 64.99...%	D+
65 - 69.99...%	C-
70 - 74.99...%	C
75 - 78.99...%	C+
79 - 81.99...%	B-
82 - 85.99...%	B
86 - 89.99...%	B+
90 - 92.99...%	A-
93 - 96.99...%	A
97 - 100%	A+

Class Projects

Four individual projects will be assigned during the term, each of which will require a substantial time commitment on your part. *Some projects will be divided into multiple parts, each with a separate deadline.* Many students find the workload in this course to be heavy.

The most common reason for not doing well on the projects is not starting them early enough. You will be given plenty of time to complete each project. However, if you wait until the last minute to start, you may not be able to finish. Plan to do some work on a project every day. Also plan to have it finished about 2 days ahead of the due date - many unexpected problems arise during programming, especially in the debugging phase. Office hours can become quite crowded as deadlines approach, making it difficult to get help. Plan for these things to happen. Your lack of starting early is not an excuse for turning in your project late, even if some unfortunate situations arise such as having your computer crash.

There are many sources of help on which you can draw. Many questions can be submitted to the course staff and your fellow classmates via Ed. These will typically be answered within the day, often more quickly during working hours. However, some types of questions cannot be answered without seeing your project. If you have detailed questions on your program, speak to a GSI or professor during office hours.

Students are also encouraged to help one another on the course concepts (but not the implementation of the projects). One of the best ways for you to make sure that you understand a concept is to explain it to someone else. Keep in mind, however, that you should not expect anyone else to do any part of your project for you. *The project that you turn in must be completely your own.*

Many [campus computing sites](#) have consultants who are available to help you at the site. They are fine sources of help with questions regarding the computers and installed software (such as Unix and the C compiler). However, they are not likely to be able to help you with questions about computer programming, the C language, or specific errors in your program.

Turning in Projects

Projects are due at **11:55 pm on the due date**. You will be submitting your projects electronically by going to [autograder.io](#). Your projects will be graded automatically using an autograder program. You are allowed to submit your programs as frequently as you wish. However, to deter you from using the autograder as a debugger, you will receive autograder feedback only **THREE TIMES** a day. That is, only the first 3 submissions will be graded and returned to you with your score on any given day. Additional submissions beyond the initial 3 will be graded and recorded, but you will not receive any feedback about them. For each project, your final score will be derived from your best submission to the autograder.

Project Late Days

Sometimes unexpected problems make it difficult to get a project in on time. For this reason, each person will have a total of **4 free late days** to be used for projects throughout the semester.

If a project is due Monday 11:55 PM, and a person submits it any time before Tuesday 11:55 PM it will use up one late day. Similarly, if they submit it after Tuesday 11:55 PM but before Wednesday 11:55 PM it will use up two late days. The same logic applies for using 3 or more late days.

Note that late days are shared across all projects, and in the case of project 1 and 2, each project part. For example, if Project 1s and 1m are due on the same day, and you use up one late day for Project 1s,

and then on the same (late) day submit for Project 1m, you will end up using two late days. (***Extension requests should be submitted within 24 hours of project due date***).

These should not be used simply to start later on a project or because you are having difficulty completing the project. Thus, please plan your work accordingly so that you won't need to use any of the late days unless there is a personal emergency. Students will generally be directed to use late days for things like illnesses or unexpected emergencies. If you are having trouble understanding the material or designing a program, please come to office hours for help right away.

Project Grading

The projects will be graded primarily for correctness (doing all the required tasks, simulating at the correct hardware level, and giving correct results). The projects will be compiled and tested on the course autograder. There are no graded private test cases so the highest score you see on the autograder is the score you will receive. A small portion of a project's grade is submitting test cases to expose bugs in an instructor made "buggy" solution. Exposing these bugs will help you debug your own code and as well give you points towards the project.

Academic Integrity

We encourage collaboration in EECS 370, especially on concepts, tools, specifications, and strategies.

All work you submit must be your own. Collaboration must not result in code that is identifiably similar to other solutions, past or present.

Encouraged Collaboration	Unacceptable collaboration
Sharing high-level design strategies, e.g., helper function organization or data structure choices	Walking through an important piece of code step-by-step, sharing pseudocode, sharing comments in your code
Helping others understand the spec or project nuances	Providing your code as a reference, or looking at solutions found online as a reference
Helping someone debug	Debugging someone's code for them

Explaining a compiler error to someone	Fixing a compiler error for someone
Discussing test strategies	Sharing test code to verify someone's design, even if test cases are not submitted
Brainstorming edge cases for testing	Discussing specifics about what tests exposed instructor bugs on the autograder
Using starter code provided with a project or based on examples shown in lecture	<p>Copying code in whole or in part, even if the code is modified</p> <p>Writing original code for someone else, or paying someone to write your project</p>
Looking at someone else's code to understand concepts or help someone debug	Sharing your code in a way that could be copied, e.g., sending code over email or taking a picture of code

We consider getting help from AI tools like ChatGPT the same as getting help from humans. You are encouraged to use these tools to ask questions and improve your conceptual understanding (while verifying any answers it gives), but all work you submit (e.g. code or homework solutions) must be written solely by you or approved group members.

You are still responsible for following these rules even after finishing the course.

If you are unsure about what constitutes an honor code violation, please contact the course staff with questions.

For those retaking the course, you are allowed to resubmit work from a previous semester as long as it is entirely your own work. Note that code suspected to be an honor code violation may still be reported, even if the same code was submitted in a previous semester without issue. It is your responsibility to ensure submitted code is entirely your work.

Homework

There will be **four** individual homework assignments during the semester, and there is NO late day for homework assignments. Your homework assignments will be posted on this website. Homework assignments are due at **11:55 pm on the due date**. Note that help may not be available after working hours. Submissions turned in by 1:00am the following day will be accepted but with a 5% penalty. Assignments turned in between 1:00am and 11:55pm (about 24 hours after the due date) will get a 30% penalty, and any submissions made after this time will not be accepted. Students can re-submit as much as they would like until the deadline. Only the last submission will be graded, and homeworks are graded for accuracy. The files can be typed up or handwritten and then scanned. **Do note that you can drop one homework score.**

You may not post homework answers on Ed or place them in any other public forum (a webpage for example) or share them with others. You may discuss your solution methods with other students, but the solutions you submit must be your own. Note that homework constitutes a fairly small factor in your final grade. Their main purpose is to give you examples of the type of questions that you will find in the course exams.

Exams

There will be **two** in-person exams this semester. You are expected to take the exams at the scheduled times. If you do not take an exam without verifying a documented medical or personal emergency causing you to miss an exam, you will receive a zero for that exam. If you anticipate conflicts with the exam time, **declare your conflict by the given due date**. The exam dates are announced at the beginning of the semester so you can avoid scheduling job interviews or other commitments on exam days. We may approve alternate exam requests for students with a valid, documented conflict with a required activity in another course, or university-affiliated teams.

If you need to request any special accommodation during any exam, please contact the Services for Students with Disabilities office. We may not be able to accommodate SSD requests submitted less than 1 week before the exam. For last-minute emergency accommodations (e.g., documented illness), please contact the staff at eeecs370admin@umich.edu.

Textbook

We recommend the following textbook for reference, but it is not required:

Computer Organization and Design, Patterson and Hennessy, ARM Edition.

Computers

You may develop on any machine that runs gcc. **However, we will only support programs which run correctly on a CAEN machine.** If you are not an engineering student (or declared CS-major), we recommend you to get a CAEN account. The autograder should work for submissions from ITCS Linux workstations, but we cannot guarantee this.

Finally, you are responsible for reading your umich e-mail inbox and instructor notes on Ed; class announcements and other student communications may be sent there.

Course Etiquette

All course participants (course staff and students) are expected to behave professionally and respectfully. Unprofessional or disrespectful behavior on the part of any course participant will not be tolerated, and we highly encourage all incidents to be reported to a member of course staff.

- All interactions must be respectful and professional, in and out of class, on Ed, in labs, in lectures
- All questions are welcomed. At any level, questions are always appreciated and always answered to the best of our abilities.
- Feedback should be constructive, as all feedback can always be expressed constructively.
- All are encouraged to participate in class interactions; everyone's experience in this class matters.
- Class decisions strive to be equitable. Class decisions will be made based on what is most fair for the majority of EECS 370 students
- All are accountable. Please report any problems to staff or one of the professors, including if the issue is with a staff member or another professor

Right to Revise

The course staff reserve the right to make changes to the syllabus at any time, as they see fit. When a revision occurs, it will be announced through Ed, and it is your responsibility to be informed of such.