CompSci 335
Inside the Box: How Computers Work
Course Syllabus
Fall, 2020: MW 4:00 - 5:15 ET, Online

Chip Weems
Office: CS-342
E-mail: weems@cs.umass.edu
Phone: 413-545-3163 (or email request for call)

TA: Devdhar Patel, devdharpatel@cs.umass.edu, Tu, Th 2 - 3

Optional Textbook: Computer Organization and Architecture, Themes and Variations, Alan Clements, Cengage Learning, 2014 (very helpful for filling in beyond the lectures)

Required Hardware: Because of our remote learning situation, it is not possible to loan out microcontroller kits this semester. Thus, you will need to purchase a kit. I have arranged with Circuit Specialists to pre-package the kit components. We will start using the kit by about the 6th class, and prior to that you will need to follow a set of online videos to assemble and test the components. Be sure to order it early. Changes have been made to the kit to reduce its cost to $78. Direct order link: https://www.circuitspecialists.com/umass_compsci_335_lab_kit.html

To use the development environment for the microcontroller you will need a computer that can access the internet, run a web browser, and that has a USB-A port or adapter. You will also need to be able to run Java (for Logisim), so a stock Chromebook will not be sufficient.

Course Objectives: As computer scientists, we tend to program using abstractions significantly removed from the hardware on which our code executes. Yet there are many cases where misunderstanding how the hardware works leads to poor performance, errors, security vulnerabilities, and other modes of failure. Software can’t completely hide physics! A deeper understanding of the underlying mechanisms of computers also helps us distinguish real technological advances from marketing hype. In this course, we pull back the covers to see how machines operate at the lowest level, and examine the electronic, mechanical, and physical principles behind the operations. We will also explore enhancements to machine architecture beyond the basics.

Course Notes: Available on Moodle and at:
https://people.cs.umass.edu/~weems/homepage/courses/cmpsci-335.html

I keep a blog there with an outline of what we cover in class, plus the slides. If you miss a class, it is important to check Moodle or the blog for updates. I do my best to update them as soon as possible, but other commitments sometimes cause me to fall behind for a few days.

Grading: In class exercises and participation in class: 50% Logic labs: 16%
Assembly labs: 12% Project proposal: 2%
Term project: 16% Project Presentation: 4%

Grade Scale:
A: 93%, A-: 89%, B+: 85%, B: 80%, B-: 77%, C+: 74%, C: 70%, C-: 67%, D+: 63%, D: 60%

These percentages are estimates, and I reserve the right to make minor changes as necessary.
In class exercises and attendance/participation in class: We will meet synchronously via Zoom during the scheduled class periods. Most periods will have an in-class exercise to prepare for or solidify the material presented in the lecture portion. Those may include building a circuit with the logic simulator or prototyping board, demonstrating something with the microcontroller, engaging in a small group discussion, and other activities. The first two periods, during add/drop are not counted. The remaining 25 will be 2% each. Many of these cannot be made up if you miss class, so attendance is important. If you have an unavoidable conflict, or illness, let me know in advance and I will try to work something out. Some of the exercises will be things that can be made up outside of class, so do ask if that's possible. Don't just assume that it's not possible. 2% seems like a small amount, but it can add up quickly to a significant fraction of your grade. This work takes the place of exams this semester. There are no exams, but the final exam period will be used for project demonstrations.

Submission Methods:
Video/photo submissions: Some of the in-class exercises will involve construction of a circuit and/or executing code on the microcontroller. Evidence of having participated in the exercise will be in the form of submitting a short video of the circuit operating, or a photo of something on the prototyping board. Thus, you will need a way to shoot videos and photos of your work (a cell phone is fine). Be sure your prototyping board has your name on it and is visible in the imagery. Videos should be submitted as MP4 via a Voice Thread activity in Moodle. Images should be submitted as jpeg or png files, as a normal file submission in Moodle.

Screenshot submissions: The logic labs and some of the in-class exercises will be done with software packages (Logisim, MBED environment), and part of demonstrating completion will be capturing a screen image. Make sure that you know how to do this for your operating system. These images should also be in jpeg or png form.

Discussion submissions: Some of the in-class work will involve discussions and problem solving by randomly assigned breakout groups. The discussions should be recorded on shared Google Docs. The process will be as follows: Upon entering the breakout room, one person will volunteer to create the doc. They will then get the sharing link (with edit permission) and post it in the Zoom chat. Everyone else in the room opens the doc, and enters their name at the top. As the discussion proceeds, everyone enters their ideas, using their initials or name to identify their contributions. When the room closes, the person who created the doc downloads it as a pdf and submits that via Moodle.

Other activities: These will be normal Moodle submissions, such as worksheets, code files, etc.

Lab Homework
Due Dates: Homework is generally due at the start of class one week after it is handed out.
Late Policy: Much of the homework is preparation for work we'll be doing in class, so it is important to keep up. It is better to turn in incomplete work and get partial credit than to fall behind and start missing assignments. If you know you have a specific time conflict, make arrangements with me in advance for a separate assignment for late submission. In some cases, feedback will be given when the homework shows a lack of understanding, and a makeup assignment will be given — the goal of the homework is for you to go through an experience that helps you to learn key concepts. It is not focused on producing a product.
Collaboration: If someone asks you for help on the lab homework, try to take the role of a teacher and help them find the solution on their own; don't just give them an answer. If it is
clear that people are submitting answers that are merely copies of each other, the credit for the one solution will be divided among the copies. Of course, the TA and I are here to help, and welcome questions, so you can always suggest that someone contact us.

Grading Tip: The homework involves using tools to gain a deeper understanding of how a computer works at the level of the digital logic, and the native instruction set. These are more about having an individual experience, than getting the right answer (although the goal of the experience is for you to get something working).

I typically lean toward generosity for partial credit, but the TA and I can do that only when you’ve made an attempt. If you leave a blank on an in-class exercise or don’t submit a lab, then it can’t be given any credit. **Always try!**

**Lab Homework Summaries**

**Logic Labs**: There are four digital logic labs, each of which has an associated handout that walks you through a step-by-step process of creating a series of circuits in Logisim. The handout asks you to experiment with the circuits and answer questions at various stages in the construction process and submit screen shots. They are briefly summarized here:

Logic lab 1: Use Logisim to simulate and explore simple combinational Boolean logic circuits: AND, OR, NOT, NAND, NOR, XOR, discover DeMorgan’s Law. (1 hour)

Logic Lab 2: Use Logisim to build a full adder, then a multibit adder/subtracter. (1 hour)

Logic Lab 3: Use Logisim to build an RS flip flop, a D flip flop, a register, and then a simple processor datapath with multiple functions. (2 hours)

Logic Lab 4: Extend the data path to include a program counter, memory, and branch logic. By this point you will have seen how a computer works to execute instructions. (2 hours)

Logisim simulator download info can be found here: [www.cburch.com/logisim](http://www.cburch.com/logisim)

**Assembly Programming Labs**: Three projects that are designed to get you up to speed on mixed C and Assembly programming for the ARM processor using the MBED cloud-based development environment. They give you a sense of how software interacts with hardware.

Programming Lab 1: Build an assembly language delay loop subroutine that can be called from C or assembly language, using the C calling convention, to pause for a specified amount of time. (2 hours)

Programming Lab 2: Use assembly to get the lights on the microcontroller to illuminate to show the low order 5 bits of two values passed in by a C program for a time that is passed as a third parameter (I/O port driver and calling the subroutine from Lab 1). (3 hours)

Programming Lab 3: Sort a table of 32 numbers passed from C using assembly language, and display the outer and inner loop counters in the lights (nested looping and calling, array access). (4 to 6 hours)

**Semester Project**

Using the the microcontroller on the prototyping board and additional components, develop a project of your choice that you think will impress the rest of the class. You may add components of your own to those on the board if you wish - there are many sources online, such as Adafruit and Sparkfun. (8 to ? hours)

The term project will be done in teams of two, although individual projects may be allowed for people with prior experience. It's helpful if the team can identify a clear division of labor, but team members should also be helping each other through challenges. This is a place where
collaboration is encouraged. Unless you are in a living arrangement where you can safely work together in person or have some other way to do a multi-microcontroller project, the expectation is that team members will run the final demonstration on just one of their microcontrollers.

You will submit a brief report describing the development process (including who did what), and what you learned from it, along with your code. It is OK to include code from the web, but you must clearly identify which code is yours and which is imported, and from where. In addition to the report, the grading criteria for the project will include the depth of learning involved (which can be shown through factors such as algorithmic complexity and/or use of different features of the microcontroller), and how well the project fulfills its goals.

There will be a proposal form to fill out for your project, so you can get feedback on it before you get too far along. Credit is given for submitting it.

The presentation will be done as a 3-minute (maximum) video, shown to the class during the final exam period. Each presentation will be evaluated in part (very small fraction) by other students in the class on the basis of its impressiveness. Mostly the evaluation is based on how well it conveys the operation of the project and the effort that went into its development. A typical video will show the project operating while a voice narration describes what is being shown and the development process. Submitting impressiveness evaluations for the presentations will be treated as a participation exercise.

**Project Teams:** When you are in a team, then your grade is partially dependent on your performance as a team. Some ways that you can help to ensure that your team does well:

Be realistic with your partner in discussing your abilities and time commitments when you divide up the project work. If possible, get together and work collaboratively, taking time to brainstorm ideas for how you will go beyond the requirements. Paired programming is a recognized software engineering technique in industry, and is known to yield higher productivity, especially on small, intensive assignments. In many cases, you'll learn more and learn it more quickly through working together directly.

Make sure you have each other’s schedules, phone numbers, e-mail addresses, etc. Establish a clear policy of when it is OK to contact one another. Let each other know when you are going to be unavailable with enough advance warning for good contingency planning.

Communicate! Communicate! Communicate! Tell each other what you are thinking. Don’t keep thoughts inside. Remember to praise each other for jobs well done. If you feel the need to criticize, use statements that start with "I think..." or "I feel..." Avoid criticism that starts with an accusatory "You did...", or "You always..." Be very clear about who is doing what - in the final project summary, I will ask you to jointly summarize what each of you contributed.

Think of yourselves as a team. Develop some team spirit. Name your team. Develop a unique style for your presentations and demos (e.g., team colors, a logo). Get to know each other as individuals. Discuss hobbies, career goals. If you each take the time to care about one another, then you’ll do what it takes to excel as a team.

**University Accommodation Statement**
The University of Massachusetts Amherst is committed to making reasonable, effective and appropriate accommodations to meet the needs of students with disabilities and help create a barrier-free campus. If you have a disability and require accommodations, please register with Disability Services to have an accommodation letter sent to your faculty. Information on services and materials for registering are also available on the [University of Massachusetts Amherst Disability Services page](http://www.umass.edu/disability).
Course Inclusiveness Statement
No matter who you are or how you define yourself you are welcome in this class. Each person here is a human being deserving of dignity and respect. My goal is to help you learn the subject matter in a way that you will find useful, and to help you have an enjoyable and empowering experience in doing so. It is important to keep in mind that we are all coming to this class with different backgrounds. For many, this is a first hardware course, while others have some prior experience. We are all here to learn together! There are no dumb questions! From time to time, I may enlist some students to help others in class. If I ask you to help, remember that we all have different modes of learning, and there is no stigma to be associated with needing assistance. Please reach out to me if you have any concerns.

University Academic Honesty Statement
Since the integrity of the academic enterprise of any institution of higher education requires honesty in scholarship and research, academic honesty is required of all students at the University of Massachusetts Amherst. Academic dishonesty is prohibited in all programs of the University. Academic dishonesty includes but is not limited to: cheating, fabrication, plagiarism, and facilitating dishonesty. Appropriate sanctions may be imposed on any student who has committed an act of academic dishonesty. Instructors should take reasonable steps to address academic misconduct. Any person who has reason to believe that a student has committed academic dishonesty should bring such information to the attention of the appropriate course instructor as soon as possible. Instances of academic dishonesty not related to a specific course should be brought to the attention of the appropriate department Head or Chair. Since students are expected to be familiar with this policy and the commonly accepted standards of academic integrity, ignorance of such standards is not normally sufficient evidence of lack of intent (http://www.umass.edu/dean_students/codeofconduct/acadhonesty/).