Syllabus

Course catalog information

Semester: Fall 2025

Course name: COMPUTER SYSTEMS I

Course ID: CS 3481

Section number: 101/102

Credit hours: 3

Contact hours: 5

Modality: face-to-face

Meeting days: Tuesdays, Thursdays

Meeting time: 9-9:50am (Section 101); 10-10:50am (Section 102)

Meeting location: 310 Anne Belk Hall

Course description: The hardware and software components of computer systems are studied. Included are those aspects of computer systems that are essential to highly skilled programmers: data representation and operations, source code implementation, processor architecture, program optimization, memory hierarchy, linking, exceptional control flow, and virtual memory. Lecture two hours, laboratory three hours.

Course prerequisites/co-requisites: Prerequisites: CS 2450 and CS 2490 with a minimum grade of "C-" (1.7) in each. Corequisite: CS 3460.

Instructor information



Instructor name: Mark Hills

Instructor office: 312L Anne Belk Hall

Instructor phone: 828-262-7159

Instructor email: hillsma@appstate.edu (mailto:hillsma@appstate.edu)

Office hours: Monday: noon - 2:00pm (before lab) Wednesday: 1:00pm - 2:00pm (before lab) Thursday: 2:00pm - 4:00pm

Statement on an accessible environment

I believe that a diverse CS community can yield a broad set of creative ideas and innovative solutions. I am committed to providing a supportive and inclusive classroom in which every member feels their ideas are valued and respected.

Course objectives

By the end of the semester, CS 3481 students will be able to:

- program in C and C++
- use the git version control system
- use the gdb debugger for finding bugs in C/C++ code
- interpret and build simple makefiles
- express the binary representation of signed and unsigned integers and explain the impact of overflow and rounding
- express the binary representation of floating point numbers and explain the issues of floating pointer overflow and rounding
- answer questions about automatic type conversions and when conversions can cause a loss of precision or overflow
- step through the machine-level instructions corresponding to simple C constructs
- apply HCL (hardware control language) descriptions in determining how machine-level instructions are being executed
- calculate the speedup obtained by pipelined architectures and explain the limitations on speedup
- explain how a pipelined architecture prevents data and control hazards

Required materials and resources

Computer Systems: A Programmer's Perspective, by Randal E. Bryant and David O'Hallaron, third edition.

The following topics/book chapters will be covered:

- C/C++ programming (links to resources can be found on asulearn main page)
- A Tour of Computer Systems (chapter 1)
- Representing and Manipulating Information (chapter 2)
- Machine-Level Representation of Programs (chapter 3)
- Processor Architecture (chapter 4)

Grades

Your grade will be determined by the following components:

- Two tests, 15% each
- Final exam, 25%
- Pre-Quizzes (outside class), 4%
- Exercises (in class), 15%
- Labs, 26%

Final grades are based on a 10 point scale with +/- given within three points of either end of the scale, i.e., 93-100, A, 90-92, A-, 87-89, B+, 83-86, B, 80-82, B-, etc. If your grade falls within this range, you are guaranteed to get at least the grade indicated. Grades may or may not be curved at the end of the semester. The grade of I (incomplete) will only be given in rare circumstances.

Lecture Attendance

It is the policy of Appalachian State University that class attendance is an important part of a student's educational experience. Students are expected to attend every meeting of their classes. However, I do understand that life happens. Attendance will be taken every lecture period. If we are doing an exercise that day, this will be through collecting the exercise at the end of class. If we aren't doing an exercise that day, this will be by means of each student writing their name on an attendance sheet before leaving class. If you do not hand in an exercise or write your name on the attendance sheet, then you will be marked absent. **You can miss up to 4 class meetings with an unexcused absence (see information on excused absences below) without penalty. Each unexcused absence after 4 will result in a 5 point deduction to your final grade in the class.** You will receive a 0 on any exercises given during a missed classes, but your 2 lowest exercise grades will be dropped.

For information about excused absences, see the University policies on attendance here (https://academicaffairs.appstate.edu/sites/academicaffairs.appstate.edu/files/printable_pdf_of_the_attendance_policy.pdf). Beyond the situations mentioned there, I allow excused absences for school related activities (e.g., you may have to be gone because you are traveling for a school club) and being sick enough to go to the doctor (when you do, get a note). Please be proactive and contact me in advance if possible.

Assignment details

(Pre) Quizzes

You are expected to come to class prepared by watching screen casts and/or going over the slides and/or reading the textbook, and completing an on-line quiz. I will not spend time in class covering concepts that should already be understood by all of the students in the class and instead cover only that material which students typically have difficulty with. It is up to you to ask questions covering quiz topics that you did not understand if I don't cover the material in class.

Exercises

Class time will be spent working on an exercise. These will be given nearly every class period. You will work on the exercise and then we will go over the answers as a class. Sometimes the exercises will be preceded by a "mini-lecture". Often, a "mini-lecture" will be given while going over the exercise questions. Taking notes will be helpful to you when you later review the material. If you attend class and spend your time working on the exercise (as opposed to looking at your phone, working on something else, etc.) then you will receive full credit on the exercise.

None of the exercises can be made up if you miss class. However, two exercise grades will be dropped.

Lab

The goal of the project is to produce a program that simulates the 64-bit Y86 architecture described in the textbook. This simulator is called YESS64 (Y Eighty-Six Simulator, 64 bit architecture). Due to the size of the project, you will be working in teams of 2 students (one team of 1 if we have an odd number of students). Each lab will direct you in producing a piece of the overall project. You will receive grades on individual labs, as well as a grade on the project resulting from the cumulative lab work.

The lab is worth 26% of your grade. Your instructor will give you a grade on each lab. All lab grades together except the last will be worth 6%. The final lab (since it is a culmination of all earlier labs) is worth 20%.

Note: labs are due at midnight, but I won't grab them for grading until the morning which gives you about a 9 hour grace period. Labs will not be accepted late. I will grade whatever you have pushed to GitHub by the time I download them from GitHub for grading.

Lab Attendance: Attendance will be taken with an attendance sheet, the same as in the lecture sessions for the class. If you miss lab, you will receive a 0 on that lab assignment unless you have already completed the lab. You can make up the missed lab by attending the other lab session that week or, if that is not possible, setting up a time with me to work on the lab (either during or outside of my office hours). Send me an email to make arrangements to make up the missed lab time. Note that this should not be a routine occurrence, I'm happy to help during my office hours but they are not a replacement for the dedicated lab time.

Tests

Two tests will be given during the semester. The first test will be on Wednesday, October 15th. The second test will be on Monday, November 24th. These will not be cumulative.

Final Exam

A cumulative final exam will be given on Friday, December 5th, at 2pm in Anne Belk 116.

General course policies

My teaching philosophy

I aim to provide you with lots of learning opportunities ... slides, screencasts, pre-quizzes, exercises (in class), mini-lectures, need more practice exercises (out of class). It is up to you to decide to use those opportunities.

The pre-quizzes are very low stakes. They are provided to encourage you to engage with the material before class. The research indicates that engaging with the material for as little as 10 minutes before a class session enables better learning during the class session.

Some of you would probably prefer that I simply lecture during a class session rather that employ more active learning techniques. But again, the research shows that traditional lecture techniques are not as effective and students will zone out in as little as 10 minutes. Providing an active learning environment is more work for me, but it better enables you to learn, which is ultimately my goal (and should be yours).

I do not believe in testing students on material that wasn't introduced to them via the activities of the class. In general, I look at the exercises when making up tests. However, I almost never use problems identical to ones I've given you. You will need to understand how to solve the problems so you can apply that same technique to similar problems.

Cheating

Don't do it! The lab assignment has been around long enough so that copies could likely be obtained by anyone lacking in integrity. What's wrong with copying code? 1) If you get caught, I'll give you a 0 on the project (26% of your grade). 2) If you get caught, I will have a hard time looking at you in the same light again (as a student that works hard and wants to learn). When a prospective employer contacts me, you don't want that cheating experience to be floating around in my head. 3) You won't become a better software developer/problem solver from copying other people's code. 4) You won't experience the joy that comes from completing a hard project honestly.
Instead, if you are having difficulty with the programming, seek help. If you start feeling desperate, come talk to me about it. I will be happy to help!
It can be difficult to determine **what counts as cheating** and what does not. Here are some guidelines:
1) Showing someone else your code counts as cheating (or obtaining code from someone else).

2) Explaining how to solve a problem *without* using code in your explanation does **not** count as cheating.
3) Helping someone interpret a compilation error does **not** count as cheating.
4) Looking for and/or using solutions online does count as cheating.
5) Using AI tools to generate solutions or partial solutions does count as cheating.
6) Looking for how to correctly call a function does **not** count as cheating. For example, you will want to take a look at online resources such as http://www.cplusplus.com (http://www.cplusplus.com/doc/tutorial/) to examine C++ libraries. (Don't be afraid to try things out, but also don't just guess at the syntax.) **I will not tolerate and am very upset by dishonesty. You and I both do not want you to be in my office crying at the end of the semester.**

AI Policy

AI coding tools have rapidly improved. While powerful, they don't always produce working code. Worse yet, if you rely on them, you are artificially capping your knowledge -- you won't know enough to know whether the code they produce is correct, or maybe to even understand it when it is.

You **are** allowed to use AI as an assistant to ask questions about the course material. A good example is: "Please explain twos-complement for binary numbers.". You **are not** allowed to use AI to generate partial or complete code for your lab assignments. This is mentioned above as point 5. Again, the point of the course is to learn the material, and that may require some digging and experimentation.

How to Succeed in CS 3481

1) Learn the material incrementally. The quizzes, exercises, and tests are designed to allow you to master the material incrementally. If you take the quizzes seriously, you'll be better prepared for working on exercises. If you master the material covered by the quizzes and exercises, you'll be prepared for the tests. There should be no reason to cram for tests because you'll be learning the material incrementally.
2) Find some people to study with. Numerous studies indicate that students earn better grades when they have peers that can help them understand the material. I have no problem with you taking the quizzes with someone else as long as you've both prepared beforehand and are both contributing to coming up with the answers.
3) Ask questions. If you don't feel comfortable asking questions in class, come by my office or send me an email. Also, feel free to ask using the Course Q&A forum.
4) Keep an open mind. Don't close your mind to learning because you think systems doesn't apply to your future. You'll be surprised about how much systems helps you mature as a computer scientist. No one knows the specifics of your professional career. What we do know is that your professional career will require solving problems and hard classes are great for helping you develop problem solving skills.5) Stop telling yourself that you are bad at tests. I have yet to have a student who is bad at tests. What I have encountered are students who are don't prepare properly for tests.

Be here. Be present. Be considerate. Be honest. Work hard.

Yes, it's a hard class, but that doesn't mean you can't succeed. You can do many hard things given the right attitude.

Withdraw dates

Withdraw no academic penalty: Friday, August 22, 2025

Withdraw using a career drop: Wednesday, October 22, 2025

Learn more about how to withdraw in the Withdrawal Policy (https://registrar.appstate.edu/resources/policies/academic-policies/withdrawal-policy)

Syllabi Policy and Statement Information (https://academicaffairs.appstate.edu/resources-forms/syllabi-policy-and-statement-information)

Additional Syllabi Policy and Statement Information (https://academicaffairs.appstate.edu/resources-forms/optional-syllabi-policy-and-statement-information)