CS 2216: Media Device Architecture
Summer 2026 GT-Europe

Instructor

Aaron Hansen
- Office Hours: TBD
- Office Location: TBD
- Email: aaron.hansen@gatech.edu
  - Please begin email subject lines with "CS2261:".

Instructor Office Hours and TA information will be also provided on Canvas.

Lecture

Day 1 and 3 (typically Mon./Wed.) 13:30 - 15:25 (1:30 pm - 3:25 pm)

Course Description

Controlling the interface between hardware and software in media devices. Machine-level programming (e.g., in C) to create graphics, generate sound, and support user interaction.

Purpose and Goals

This course is intended to greatly demystify what a computer is and how exactly it works. By the end of this course, the 0s and 1s of computing should no longer be an abstract mystery that exists "somewhere" in the computer. This course is intended to have students understand how hardware and software architecture affects what is easy vs hard to program on a given machine. It should provide an understanding of programming at the level of shuffling bits and bytes, as well as at the “higher level” and in between.

To those ends, by the end of the course students should:

- Understand data representation. Be able to convert numbers between various representations: Binary, octal, decimal and hexadecimal.
- Be able to identify the component parts of the Von Neumann Model of computer and be able to explain the purpose of each component.
- Be able to use utilities for conversion of image and sound files into C data structures
- Be able to write C programs that are hundreds to thousands of lines long that manipulate device hardware and perform some logical functions such as creating a playable video game.
- Be able to design, write and debug code to perform operations such as:
  - Displaying images in multiple formats

    - Creating simple animations used in bit-mapped games
    - Using page flipping / double buffering for smooth animation
    - Using Dynamic Memory Access (DMA) to speed up animations
    - Effectively use indexed color, tiles and sprites.
    - Starting, stopping and looping sounds
    - Reading button inputs
    - Maintaining and changing program state
    - Servicing hardware interrupts

Note: This course is not designed to train students to be “C developers,” although some may take away information useful in such an endeavor.

Grading

- Attendance: 5%
- Participation Exercises: 5%
- Homeworks / Milestones: 25%
- Exams: 40%
- Final Project (in lieu of Final Exam): 25%

Grade Cutoffs: A: 90, B: 80, C: 70, D: 60 (grades will not be rounded)

Homework / Milestones

Homeworks for this course are typically larger assignments that span a week or two. They are often open-ended with specific technical requirements. Each homework submission must be completely original (meaning completely new code must be written, by you, for each submission), unless otherwise specified. There will be roughly four (4) homework assignments.

Milestones come at the end of the semester as work on the Final Project begins. Milestones are generally similarly scoped to homeworks, but each Milestone should build upon the previous one as you make incremental progress towards completion of the Final Project. Milestone grading criteria include specific "follow-through" points, based on how well you responded to direction given by your TA on your previous Milestone. There will be three (3) Milestones, barring any unforeseen schedule changes. One or more Milestones may be required to undergo a demo process where you answer questions about how your submitted code accomplishes project behavior -- an inability to adequately explain your own code during this demo will lead to a reduction in your score on the assignment.

Late Homework/Milestones will be accepted up to 48 hours late with a 5% per-day deduction to the resulting grade.

Exams

There will be two (2) in-class written midterm exams that will be given during the officially assigned lecture time. Each will cover roughly five (5) weeks worth of course material. Each exam will be equally weighted when calculating your exam average.

Final Project

The Final Project will be the culmination of roughly three weeks of work at the end of the semester. It is an individually imagined, designed, and developed game. Demoing the project will be required, but will be worth very little of the final project grade. Exact details and requirements will be announced over the course of the semester.

No late Final Projects will be accepted.

Attendance

Attendance will be pseudo-randomly taken seven (7) times during the semester. If you are not present on at least five (5) of the days, you will lose one percentage point on your final grade per day below that threshold (e.g. if you are only present on 3 of the 7 attendance checks, you will lose 2% from this category).

Participation Exercises / "Daily" Mini-Quizzes

For each lecture day, there will be an associated mini-quiz on Canvas. You are encouraged to complete each mini-quiz the day of the associated lecture, but the mini-quizzes will generally not be due until Friday of each week (allowing for some individual scheduling flexibility). These mini-quizzes are low-stakes and you will have unlimited attempts to complete them (up to the due date).

Mini-Quizzes will not be eligible for make-up and will not be accepted late.

Make-up Policy

If you are sick and still feel up to attending, please wear a mask to protect those around you. If you are too ill to attend class, please email the instructor ***in advance*** of the absence, particularly if there is an Exam that day.

Students are responsible for all material covered in class, including changes in schedules announced during class or on Canvas. Make-up assignments are not permitted, except due to an officially excused absence or extreme circumstance. **Students missing an assignment**

(including an exam!) without an excused absence will receive a grade of zero on that assignment. All non-emergency exceptions to the late policy must be approved in advance by the Head TA or the instructor.

If there is a problem submitting an assignment (including due to a diagnosed illness), email the course instructor and the Head TA as soon as possible! We are human beings and will try to be reasonable and fair, but you must make every effort to notify us promptly.

Back up your source code early and often. An external git repository with a local working copy (via a private repo on Bitbucket, GitHub, etc.) is a good option. Even then, you might want to make sure your project folder is automatically backed up to some off-site/offline location. If you experience catastrophic hardware failure, we will work with you while your equipment is being repaired. However no extensions will be granted due to your lack of backups.

If illness or other life events will cause you to be absent for >24 hours (and/or if you are absent for an Exam), please work with GTE administration to validate your absence and facilitate possible accommodations:

- paul.voss@ece.gatech.edu,
- laila.abou_dahab@georgiatech-metz.fr, or
- sarah.rucho-ghezal@georgiatech-metz.fr

Schedule

A "Living Schedule" document will be made available on Canvas. The specific topics/dates are subject to potential changes, but the provided schedule will be kept up-to-date as changes are announced. Please make use of this schedule as you make plans around your required participation in this course. The one assignment due date that cannot be changed is the Final Project Due date (this can only be changed by the registrar, and any change will be communicated as soon as possible).

Academic Integrity and Collaboration

Every student is expected to read, understand and abide by the Georgia Tech Academic Honor Code. Academic misconduct is taken very seriously in this class. Your homework assignments may be evaluated via demo or code review. During this evaluation, you will be expected to be able to explain every aspect of your submission. You are expressly forbidden to supply a copy of your homework to other students via electronic means. If you supply an electronic copy of your homework to another student, and they are charged with an academic integrity violation, you will also be charged. Collaboration with other students currently in this class is an important learning method. The following explanation will help you understand collaboration. Students may only collaborate with fellow students currently taking CS 2316, the TAs, and the

instructor. Collaboration means talking through problems, assisting with debugging, explaining a concept, etc. You should not exchange code or write code for others. **Each individual assignment must be coded by you.** Your submission must not be substantially similar to another student's submission. Collaboration at a reasonable level will not result in substantially similar code. **Students that turn in submissions that are not fundamentally unique will receive a zero (0) and will be referred to the Office of Student Integrity (OSI).**

We expect academic honor and integrity from students. Please study and follow the academic honor code of Georgia Tech: http://www.honor.gatech.edu/content/2/the-honor-code. You may collaborate on homework assignments and daily work, but your submissions must be your own. You may not collaborate on exams. We will turn students into the Office of Student Integrity if we suspect that the honor code has been violated.

AI Tools Clarification:
Using ChatGPT, GitHub Copilot, or any other code generation tool is forbidden in this course. You can ask ChatGPT, Copilot, etc. conceptual or technical questions (as you would the instructor or a TA), but you should be careful of relying on its response as accurate -- its knowledge of the GameBoy Advance is incomplete. You need to have personally written every line of code in your assignment submissions.

Regrade Policy

To contest any grade you must contact the instructor **within one week of the assignment's original return date**. The original return date is the date the exam was first made available for students to pick up or the grade was posted on Canvas. A week after the grade is available, regrade requests will no longer be accepted.

Prerequisites

- Undergraduate Semester level CS 1331 Minimum Grade of C

Course Materials

Programming Language and IDE

The language used in this class is C. We will edit our code using a text editor such as Visual Studio Code. Instructions to set up your environment will be available on **Canvas**.

Canvas / Gradescope

All course information and resources can be found on Canvas. This includes: Syllabus, Assignments, Submissions, Announcements, Grades & Feedback, Resources, etc.

The code from each lecture will be posted on Canvas under the Files tab by the end of the following day.

Exams will most likely be scanned and grades returned via Gradescope (accessible via Canvas). Gradescope is also where regrade requests for exams will be submitted / processed.

Textbooks / Resources

- Required Readings provided on Canvas
- *Recommended* resources:
    - TONC - https://www.coranac.com/tonc/text/
        - This resource is very relevant to this course!
    - The C Programming Language - Kernighan and Ritchie (a.k.a., "K&R")
        - The first de facto standard of C -- no need to run out and buy it.
    - GBATek - https://problemkaputt.de/gbatek.htm
        - Very technical explanations of GBA hardware -- useful only late in the course.

Internet Connectivity and Computer Ownership Expectations

Your computer must meet the requirements laid out by the Georgia Tech computer ownership policy which may be found here: https://sco.gatech.edu/hardware-requirements/.