

Course Syllabus

[Jump to Today](#)

 [Edit](#)

Course Description

This course covers modern computer architecture, including branch prediction, out-of-order instruction execution, cache optimizations, multi-level caches, memory and storage, cache coherence and consistency, and multi- and many-core processors.

Professor and TAs

Professor

Milos Prvulovic

Office hours: Wednesdays 12:05pm to 12:55pm in Atlanta (US Eastern time), we will be using BlueJeans, and you can use this [link](https://bluejeans.com/307250233) [_ \(https://bluejeans.com/307250233\)](https://bluejeans.com/307250233) to join the meetings.

TAs

TBD

Prerequisites

Undergraduate computer architecture course that covers basic computer organization; working knowledge of topics such as instruction sets, pipelining, etc. For the course project, you will also need to be familiar with C/C++, Linux, and be comfortable making modifications to large programs.

If you answer “no” to any of the following questions, it may be beneficial to refresh your knowledge of the prerequisite material prior to taking CS 6290:

- Have you taken a computer organization course before?
- Are you familiar with at least one RISC instruction set and would you feel comfortable reading and writing small assembler programs?
- Are you familiar with basic computer architecture concepts, such as pipelines and caches?
- Are you familiar with C/C++ and would you be comfortable writing and/or modifying 100+ lines of code in a program that has over 100,000 lines of code?
- Are you comfortable with, or even excited about, learning about how real processors work and using simulation to see how changes in processor design affect its performance?
- Have you successfully completed the HPCA0 "course" at Udacity?

Textbook

There are no required readings. When appropriate, additional class materials will be available as instructor notes that are associated with the video lectures. Although we do not require, and do not officially recommend, a textbook, a useful textbook for this course is “Computer Architecture: A

Quantitative Approach” by John L. Hennessy and David A. Patterson. A recent edition should work, but editions 1-4 put less emphasis on multi-core topics than our course does.

Grading

The grade is determined by your performance on projects and exams. You will receive these grades through Canvas. The projects and exams will count toward the final grade as follows:

- Projects (50% of overall grade): You will be given four projects, each requiring more work than the previous one. Each project is to be completed individually or in two-student teams, as specified in each project assignment.
 - Project 0 (5% of overall grade)
 - Project 1 (10% of overall grade)
 - Project 2 (15% of overall grade)
 - Project 3 (20% of overall grade)
- Exams (50% of overall grade)
 - Midterm (20% of overall grade)
 - Final (30% of overall grade), it **does** include questions about material covered in the Midterm

The plan is to assign final (letter) grades based on your total score, with 90% and above earning an A, 80% and above earning a B, etc. If this results in too few As, we may decide to lower the thresholds somewhat, or to use some sort of a curve - the final decision whether and what to do in this regard is up to the instructor.

There will be **no make-up assignments**, so if you need a particular grade plan to perform accordingly on projects and exams. Once a homework, project, or exam is over and graded, the only way the score on that assignment or exam will be changed is if a legitimate mistake in grading has been made. Due to the large number of students in this class, assignment and exam re-grades can only be requested during 14 days that follow the release of scores from that assignment/exam. When requesting a re-grade, keep in mind **that the entire submitted project/exam will be regraded**, so a request for a regrade may result in a net loss of points.

The grade in this class will be based solely on demonstrated performance. No grade will ever be changed because the student **needs** a better grade to stay in the program, to keep a fellowship, to get a job, or any other reason. If you believe you need some particular grade in this class, the only way to get that grade is to earn it on projects and exams.

Emergencies and Late Policy

No late assignments or exams will be accepted unless we are advised to do so by the Dean of Students. Please contact the office of the Dean of Students with health emergencies, family emergencies, personal disabilities, or other significant events. The Dean’s office is equipped to verify these exceptions better than us, and provides a level of uniformity across courses on how emergencies are handled.

Academic Integrity

All Georgia Tech students are expected to uphold the [Georgia Tech Honor Code](http://osi.gatech.edu/content/honor-code) (<http://osi.gatech.edu/content/honor-code>). **You should read it (including the Graduate Addendum)!** We take cheating **very** seriously. Note that all Georgia Tech faculty (including the instructor for this course) are **required** to report cases of academic dishonesty to the Dean of Students' office at Georgia Tech.

Class Schedule







This schedule lists important dates (exams, project release and due dates, etc.). The white-background items show the recommended schedule for completing lessons. Of course you can complete lessons at a faster pace, but projects and exams are timed assuming that you will take the lectures according to the provided schedule - so do not fall behind!

Sep 8-11ROB

Week	Dates	Topics
1	Jan 14-15	Introduction, Metrics and Evaluation
	Jan 15	Project 0 Released
2	Jan 18	Official School Holiday
	Jan 19-22	Pipelining Review; Branch Prediction; Predication
	Jan 22	Project 1 Released
	Jan 24	Project 0 Due at midnight AOE (GMT-12)
3	Jan 25-29	ILP and Instruction Scheduling
4	Feb 1-5	ROB
	Feb 3	Project 2 Released
5	Feb 8-12	Memory Ordering
6	Feb 15-19	Compiler ILP and VLIW
	Feb 21	Project 1 Due at midnight AOE (GMT-12)
7	Feb 22-26	Cache Review and Virtual Memory
	Feb 26-28	Midterm Exam (2-hour proctored exam)
8	Mar 1-5	Advanced Caches, Memory
	Mar 3	Project 3 Released
9	Mar 8-12	Storage and Fault Tolerance
	Mar 14	Project 2 Due at midnight AOE (GMT-12)
10	Mar 15, 17-19	Multi-Processing

Week	Dates	Topics
	Mar 16	Mid-semester break from instruction
	Mar 17	Last day to drop course with "W" grade
11	Mar 22,23,25,26	Cache Coherence
	Mar 24	Mid-semester break from instruction
12	Mar 29 - Apr 2	Synchronization
13	Apr 5-9	Memory Consistency
14	Apr 12-16	Many-Core
	Apr 18	Project 3 Due at midnight AOE (GMT-12)
15	Apr 19-23	Review and Prepare for Exams
16	Apr 26-28	Final Instructional Class Days and Reading Day (Prepare for Exams)
	Apr 30-May 2	Final Exam (3-hour proctored exam)

Course Summary:

Date	Details	
Mon Jan 25, 2021	 Project 0 (https://gatech.instructure.com/courses/163742/assignments/630532)	due by 7am
Mon Feb 22, 2021	 Project 1 (https://gatech.instructure.com/courses/163742/assignments/630534)	due by 7am
Mon Mar 1, 2021	 Midterm (https://gatech.instructure.com/courses/163742/assignments/630528)	due by 9am
Mon Mar 15, 2021	 Project 2 (https://gatech.instructure.com/courses/163742/assignments/630536)	due by 8am
Mon Apr 19, 2021	 Project 3 (https://gatech.instructure.com/courses/163742/assignments/630538)	due by 8am
Mon May 3, 2021	 Final (https://gatech.instructure.com/courses/163742/assignments/630530)	due by 11am