

# CSE 6230 Syllabus - Georgia Institute of Technology

## High Performance Parallel Computing: Tools and Applications

Credits: 3.000

Tu/Th 5:00-6:15 pm @ TSRB 132

## Instructor information:

### Instructor:

Professor Helen Xu

Email: [hxu615@gatech.edu](mailto:hxu615@gatech.edu)

Office: Coda S1315

Weekly office hours: after class in the same room on Thursdays

## General course information

### Description

Achieving the full potential of modern computing platforms at all scales from cell phones and laptops up to supercomputers is imperative for meeting the ever-growing demand for computational power. Unfortunately, the computational power of modern machines comes with a great deal of complexity, such that exploiting even a fraction of a computing platform's potential requires a deep understanding of all layers of the underlying platform as well as the application at hand. Learning and taking complete advantage of modern machines is necessary to develop high-performance codes that can process greater volumes of data orders of magnitude more efficiently, which will become increasingly important as data and problem sizes continue to grow.

This course teaches hands-on practical performance engineering on high-performance computer systems, starting with single-processor multicore platforms up to large-scale distributed systems. Topics include cache-friendly algorithm analysis and implementation, parallel programming, concurrent data structures, tools for effective performance engineering, and modern high-performance computing (HPC) applications.

### Course Learning Outcomes

After completing this course, students should have the following capabilities:

- Design high-performance code. Good design requires not only good algorithms for the problem at hand but also good understanding of the underlying computer system via performance models. This course will discuss practical performance modeling of current architectures, including multi-core, coprocessor (e.g., GPUs) and many-core designs.

- Implement fast code for shared-memory machines, distributed-memory machines, and GPUs.
- Become familiar with a variety of use cases for high-performance computing from different domains.
- Present and write up their proposed work and results clearly.

## Pre- &/or Co-Requisites

- Proficiency in C/C++ programming.
- Algorithms and data structures.
  - Preferred courses:
    - Algorithms (one of CS 6550 / CSE 6140 / CSE 6220)
    - Data structures (e.g., CS 1332)
- Operating systems basics (optional but preferred)

## Required Course Materials:

There is **no required textbook**. Any required reading will be available free to GT students.

We will provide links throughout the semester to additional reading to review foundational concepts as well as more advanced treatment of the topics.

Other great resources include:

Previous versions of this course:

- [Professor Chow's lectures](#)
- [Professor Vuduc's lectures](#)
- [Professor Kannan's lectures](#)

Related/similar courses:

- [Performance Engineering Of Software Systems](#) (MIT 6.172)
- [Algorithm Engineering](#) (MIT 6.506)
- [Applications of Parallel Computers](#) (UC Berkeley, CS267)

For a comprehensive background in algorithm analysis, algorithms, and data structures, see "[Introduction to Algorithms](#)" by Cormen, Leiserson, Rivest, and Stein.

For a textbook-like resource on parallel algorithms, see "[Introduction to Parallel Algorithms](#)" by Blelloch, Dhulipala, and Sun.

# Grading Policy:

## Assignment Distribution

Assignment	Weight (%)
Pre-course survey and academic honesty	1
Homework 1: Cache-friendly programming	5
Homework 2: Shared-memory programming	10
Homework 3: Data-level parallelism	5
Homework 4: MPI	20
Homework 5: Cuda	15
Project proposal writeup + meeting	6
Project midpoint report	8
Final project writeup	15
Final project presentation	15
Total	100
Extra credit (paper reviews for guest lectures)	Up to 4

All assignments are due at 4:55pm on their due date.

## Grading Scale

Grades will be earned according to the standard scale:

A	90-100%
B	80-89%
C	70-79%
D	50-69%
F	0-49%

Grades may be curved at the end of the semester to lower the cutoff. The cutoffs will never be increased when finalizing grades, they only may be lowered (i.e., you will never get a lower

grade than the one prescribed by these cutoffs, but may get a higher one depending on the curve).

**Pass/Fail Policy:** Students taking the course on a Pass/Fail basis must secure at least 50% of the total, as prescribed by the weights above.

**Audit policy:** Students taking the course as Audit should not hand in any assignments. If they do, they will not be graded.

## Description of Graded Components

### Survey

The pre-course survey (link [here](#)) will help me to understand students' background knowledge and interests to help guide the course pace and topics. There are no right or wrong answers. The questions are also intended to help serve as a check for students to make sure they have the necessary prerequisite knowledge about the foundations of the course (e.g., C programming, algorithm analysis, etc.).

### Programming homework

The course will have 5 programming homeworks to reinforce the topics covered with hands-on implementations. They will include exercises in coding as well as exercises in using performance engineering tools. They will be as follows:

- Cache-friendly programming (HW1)
- Shared-memory programming (HW2)
- Data-level parallelism / vectorization (HW3)
- MPI (HW4)
- Cuda (HW5)

### Project proposal writeup & meeting

The first step in the final project is to gather a group of up to 3 students and propose a project.

The proposed project must

- be relevant to the topics discussed in class,
- require significant programming effort from all team members,
- be unique (i.e., two groups cannot pick the same idea), and
- be approved by the instructor.

The course staff will provide sample project topics, or you are free to choose your own.

There will be two deliverables at this stage: a writeup and a meeting with the instructor. The write-up should include

- the problem statement,
- approach,

- an evaluation plan, and
- a weekly timeline of expected milestones.

The meeting should summarize all of these aforementioned points. The instructor will give feedback on the points, which should be incorporated into the writeup.

## Project midpoint report

About halfway through the project timeline, each group will submit a midpoint report on

- their progress so far,
- a breakdown of the contribution among group members so far,
- any obstacles they encountered,
- changes to the proposed tasks / approach (if any), and
- a schedule of the remaining work to be done.

The midpoint report is intended to make sure everyone is making progress and to catch any major issues early (rather than right before the final project deadline).

## Final project writeup & presentation

At the end of the semester, each group will produce a research-paper-like report on their final project. These will include sections such as:

- Introduction
- Background
- Project contribution
- Results / evaluation
- Conclusion
- References

In addition to these sections, the writeup should include a breakdown of the contributions for each member in the group.

Just as in the proposal stage, there will be two deliverables: a writeup and an in-person presentation.

You are required to cite all sources you use in your paper. This includes both direct quotations and cases where you use someone else's ideas. "Sources" include papers, journals, conversations, anything found on the internet, and so on. Basically, if the thought did not originate from you, you should provide a source.

## Extra Credit Opportunities

For extra credit, I will provide papers as background reading for the guest lectures at the end of the semester.

Each student can submit up to 2 paper reviews, each worth up to 2% of the final grade, for a total of 4% extra credit.

These extra credit assignments will have related deadlines before the guest lectures.

## Submitting Assignments - How, Where, and When

We will use Canvas for submitting and managing assignments.

For the coding homeworks, students will submit their code via Canvas.

For the final project, students will host their code on Github as well as submitting it on Canvas.

We will provide materials on how to use Github during the semester.

All assignments are **due by 4:55pm on their corresponding deadline dates**. The late policy can be found below in “Course Policies, Expectations, and Guidelines.”

## Course policies, expectations & guidelines

### Attendance Policy

Almost all of the lectures will be in-person with the potential exception of some remote guest lectures. The in-person lectures will not be recorded. The remote lectures will be projected in the classroom. We request that students turn off their mic and video if joining remotely. We will send the online Zoom/Teams link to all registered students (please send the course staff an email if you didn't get it).

Attendance is not required but expected. The lectures are a chance to directly interact with your instructor and classmates.

### Plagiarism & Academic Integrity

Georgia Tech aims to cultivate a community based on trust, academic integrity, and honor. Students are expected to act according to the highest ethical standards. For more information on the Honor Code, please visit the [OSI website](#).

Any student suspected of cheating or plagiarizing on a quiz, exam, or assignment will be reported to the Office of Student Integrity, which will investigate the incident and identify the appropriate penalty for violations.

### Accommodations for Individuals with Disabilities

If you are a student with learning needs that require special accommodation, contact the Office of Disability Services at (404) 894-2563 or <http://disabilityservices.gatech.edu/>, as soon as

possible, to make an appointment to discuss your special needs and to obtain an accommodations letter. Please also email me as soon as possible in order to set up a time to discuss your learning needs.

## Collaboration & Group Work

For all programming assignments, students may collaborate through online or in-person discussion, but **all coding and writing should be done by each individual student**. Students who submit unattributed material will be found in violation of the Honor code (see Academic Integrity above).

For the final project components (proposal, midpoint report, final report & presentation), students may work with their project groups.

## Late Policy

All assignments are due by 4:55pm on their corresponding deadline dates. There is a free 48-hour late policy on every assignment.

## Student-Faculty Expectations Agreement

At Georgia Tech we believe that it is important to continually strive for an atmosphere of mutual respect, acknowledgement, and responsibility between faculty members and the student body. See <https://catalog.gatech.edu/rules/21/> for an articulation of some basic expectations – that you can have of me, and that I have of you. In the end, simple respect for knowledge, hard work, and cordial interactions will help build the environment we seek. In other words, I encourage you to remain committed to the ideals of Georgia Tech, while in this class.

## Student Use of Mobile Devices in the Classroom

As research on learning shows, unexpected noises and movement automatically divert and capture people's attention, which means you are affecting everyone's learning experience if your cell phone, pager, laptop, etc. makes noise or is visually distracting during class.

For this reason, I ask you to turn off your mobile devices and close your laptops during class.

## Subject to Change Statement

The syllabus and course schedule may be subject to change. Changes will be communicated via the Canvas announcement tool. It is the responsibility of students to stay current.

## Use of Artificial Intelligence Statement

We treat AI-based assistance, such as ChatGPT and Copilot, the same way we treat collaboration with other people: you are welcome to talk about your ideas and work with other people, both inside and outside the class, as well as with AI-based assistants.

However, all work you submit must be your own. You should never include in your assignment anything that was not written directly by you without proper citation (including quotation marks and in-line citation for direct quotes).

Including anything you did not write in your assignment without proper citation will be treated as an academic misconduct case. If you are unsure where the line is between collaborating with AI and copying AI, we recommend the following heuristics:

- Heuristic 1: Never hit “Copy” within your conversation with an AI assistant. You can copy your own work into your own conversation, but do not copy anything from the conversation back into your assignment.
  - Instead, use your interaction with the AI assistant as a learning experience, then let your assignment reflect your improved understanding.
- Heuristic 2: Do not have your assignment and the AI agent open at the same time. Similar to the above, use your conversation with the AI as a learning experience, then close the interaction down, open your assignment, and let your assignment reflect your revised knowledge.
  - This heuristic includes avoiding using AI directly integrated into your composition environment: just as you should not let a classmate write content or code directly into your submission, so also you should avoid using tools that directly add content to your submission.

Deviating from these heuristics does not automatically qualify as academic misconduct; however, following these heuristics essentially guarantees your collaboration will not cross the line into misconduct.

In each programming assignment writeup and project component, please include a section about how you used AI.

## Campus Resources for Students

[Success at Tech](#) is a website that links to many student resources including the [LGBTQIA Center](#), [OMED](#), [Women’s Resource Center](#)), study supports ([Tutoring and Academic Support](#), the [Communication Center](#)), mental health resources ([CARE](#)), food banks and food security resources ([STAR](#)), resources for [veterans, service members, and their families](#), and others.