CSI 702 Syllabus (Spring 2021)

High-Performance Computing

Description

This course will teach students programming techniques, tools, and debugging methods for using parallel computers to efficiently solve challenging problems in science and engineering. We take an applied approach to learning how to optimize programs serially and then by leveraging some of the most important parallel programming models today: shared memory (OpenMP), distributed memory (MPI), and GPUs (CUDA). By completing a series of mini-projects, students will develop an intuition for identifying the patterns that appear in essentially all programs that need to run fast and learn how to address them using linear algebra, graph algorithms, structured grids, etc. The course culminates in a final project where students code their own parallel program that solves a scientific problem, either from scratch or by modifying an existing code base.

- **Classroom:** *Online* (meeting details will be emailed)
- **Meeting times:** Wednesdays 7:20pm – 10:00pm
- **University holidays:**
- **Credit hours:** 3.0 credit hours
- **Prerequisites:** Competency in Linux and programming at the CSI 501 level

Instructor

Dr. Swabir Silayi

- **Email:** ssilayi@gmu.edu
- **Office hours:** 1 hour before class or online by appointment

Platforms

The course will be administered through the following online platforms:

- Zoom - https://its.gmu.edu/service/zoom/
- XSEDE - https://www.xsede.org
- Moodle - https://moodle.xsede.org
- Blackboard - https://mymasonportal.gmu.edu
- Slack - https://gmucsi702sp21.slack.com/
- Github - https://github.com

Languages and software

The course lectures and homework assignments are primarily in C and C++. GPUs will be programmed using Nvidia's CUDA C API and Thrust C++ template library.

While much of your code will be submitted and executed on remote clusters, code development can be handled locally. To help with development, you should install and use an IDE. If you have a favorite one that you can setup to work with C and C++, then that is fine. Otherwise, sign up for and request a free student license for JetBrains CLion: https://www.jetbrains.com/shop/eform/students.

The following is a list of software and libraries you will want to install on your laptop or desktop:

- JetBrains CLion: https://www.jetbrains.com/clion/
- Anaconda 2018.12: https://www.anaconda.com/download /
- Git: https://git-scm.com
- clang-format (may require full clang library): https://clang.llvm.org/docs/index.html
- GitHub Desktop (optional): https://desktop.github.com/

Materials

The course has no official textbook, but the following resources will be useful to supplement the course:

P. Pacheco, *An Introduction to Parallel Programming* (Morgan Kaufmann, Amsterdam: Boston, 2011). https://proquest.safaribooksonline.com/book/programming/9780123742605

D. Kirk and W. Hwu, *Programming Massively Parallel Processors: A Hands-on Approach*, 2nd ed. (Morgan Kaufmann, Amsterdam, 2012). https://proquest.safaribooksonline.com/9780124159921

V. Eijkhout, R. van de Geijn, and E. Chow, *Introduction to High Performance Scientific Computing* (Zenodo, 2016). https://doi.org/10.5281/zenodo.49897

V. Eijkhout, *Introduction to Scientific Programming in C++/Fortran2003* (unpublished). https://bitbucket.org/VictorEijkhout/textbook-introduction-to-scientific-progr amming

V. Eijkhout, *Parallel Programming in MPI and OpenMP* (unpublished). https://bitbucket.org/VictorEijkhout/parallel-computing-book

Policies

Contact policy

If you must, use your mason email for contacting me at ssilayi@gmu.edu. Otherwise, correspondence is to be done using the private, invite-only Slack workspace for the course. Direct messages on Slack can be used for contacting me instead of emails. My ground rules for direct messages are as follows:

- Allow up to approximately 24 hours for a response during normal hours.
- I check and respond to messages much less frequently over weekends and school holidays.
- For longer questions, I will ask you to schedule an appointment and come to office hours.

Attendance policy

Attendance will not be graded but is strongly encouraged. The course is generally structured in the "flipped classroom" format, meaning you should come to class prepared to ask questions and discuss the materials distributed via Moodle and for the homework assignments. I will also cover additional material that's relevant to the homework problems and high-performance computing as needed. Students are responsible for obtaining and understanding the material that they miss.

Illness and emergencies

It is a student's responsibility to inform me about illnesses or personal/family emergencies that will interfere with your ability to complete assignments. This must be done as soon as possible. You may be asked to provide a doctor's note if you request an extension on an assignment.

I understand that certain emotional or physical situations can impact a student's willingness to communicate what is going on and that it can take a couple days to inform me about a personal emergency or severe illness. At the same time, all students are expected to exercise personal responsibility. It is not acceptable to wait to tell me about the impacts of a personal illness or emergency until you're about to fail the course due to missing deadlines.

Late work policy

In general, failing to submit an assignment on time will result in a zero. Extensions may be granted in the case of illness or a family emergency at my discretion (see *Illness and emergencies* section), and it is the student's responsibility to inform me about these kinds of circumstances as soon as possible.

Accommodations policy

Students with disabilities who need academic accommodations, please see me and contact the Office of Disability Services (ODS) at (703) 993-2474. All academic accommodations must be arranged through the ODS: http://ods.gmu.edu/.

Grading

Breakdown

Category	Weight
Quiz	15%
Homework	35%
Final project	50%

Schema

Based on the final total score, your final grade will be determined as follows: A+ [97-100], A [93-96], A- [90-92], B+ [87-89], B [83-86], B- [80-82], C [70-79], F [<70].

Expectations

Quizzes

There is an online quiz following every video posted on Moodle that should be completed the same week that the video is assigned. Each quiz can be attempted multiple times and your grade will be your highest score.

Homework

Approximately 4 – 5 homework assignments will be assigned during the semester. The majority of these assignments are mini-projects that will require you to submit a formal report along with your code. Your submitted code must be uploaded in the locations that I specify. It will then be compiled and tested against a series of benchmarks. Code submissions that do not compile or produce output that the auto-grader does not understand will automatically receive a zero.

You **must** format your code so that it is clean and readable for all coding assignments. The clang-format tool can take care of this for you automatically.

Final project

Details on the final project will be handed out at a later date and will involve students optimizing the parallel performance of an existing program. This can involve implementing a parallel algorithm in serial code, improving an existing parallel solution, or writing a GPU version for the code. Students will complete their projects and submit their documented code and benchmarking report on April 30th. Students will then present their projects to the rest of the class during the scheduled final exam period on May 5th.

Conduct

Academic integrity

"Student members of the George Mason University community pledge not to cheat, plagiarize, steal, or lie in matters related to academic work." - Office for Academic Integrity. *2017-2018 Honor Code and Honor System.* Web. 27 Aug. 2017.

Students may discuss non-group work outside of class, however in all instances it is required that your submitted work and code are your own. Do not duplicate or paraphrase another person's material or ideas and represent them as your own. Content that comes from a resource or another student should be properly cited.

*A note on sharing or reusing code found on other Github repos or on websites like *Wikipedia* or *Stack Overflow*. It's common knowledge that researchers in both industry and academia will use search engines while writing code. Being able to search for existing solutions so that you don't "reinvent the wheel" is a useful skill. Therefore, unless I specify otherwise, you are permitted to use these resources **as long as you provide a citation**.

Exceptions to this rule are:

- For individual assignments, you cannot use another student's code (past or present).
- For group assignments, you cannot reuse another group's code (past or present).
- You are not permitted to refer to solution sets for any of the homework problems.
- For the final project, you cannot use or reference another person's optimized implementation for the same code base, a similar code base, or in a different programming language.

ANY MATERIAL THAT IS TAKEN IN WHOLE OR IN PART FROM ANOTHER SOURCE AND NOT PROPERLY CITED WILL BE TREATED AS A VIOLATION OF MASON'S ACADEMIC HONOR CODE.

Other violations of Mason's Honor Code will be treated similarly. Suspected violations will be reported to the Office of Academic Integrity. Please see the Honor Code page for details.

Decorum/discourse

Students are expected to be civil in their classroom conduct and respectful of their fellow classmates and the instructor for the duration of the course. Examples of expected behavior include, but are not limited to, not interrupting your classmates or the instructor, refraining from using devices for anything other than coursework and coming to class prepared with questions for discussion. These expectations remain in effect for all online discussions.

I will address violations of classroom decorum on a case-by-case basis and reserve the right to enact grade-based penalties for highly disruptive or repeat violations. Penalties for decorum violations cannot be negotiated or appealed.

Mason diversity statement

George Mason University promotes a living and learning environment for outstanding growth and productivity among its students, faculty and staff. An emphasis upon diversity and inclusion throughout the campus community is essential to achieve these goals. Diversity is broadly defined to include such characteristics as, but not limited to, race, ethnicity, gender, religion, age, disability, and sexual orientation. Diversity also entails different viewpoints, philosophies, and perspectives. Attention to these aspects of diversity will help promote a culture of inclusion and belonging, and an environment where diverse opinions, backgrounds and practices have the opportunity to be voiced, heard and respected.

Support services

George Mason provides Counseling and Psychological Services (CAPS) for students. Contact them at (703) 993-2380 or http://caps.gmu.edu/.

Disclaimer

The instructor reserves the right to modify this syllabus at any time during the course to improve the learning experience and classroom environment. The pacing of the course and the list of covered topics may also be altered in response to student progress.

The course objectives reflect what a student is expected to understand by the end of the course after putting in the necessary time and effort both inside and outside the classroom and completing all assignments. These outcomes are not a guarantee, and students will get more out of the course the more they put into it. Any acquired skills and knowledge can fade over time if not reviewed or practiced after the course concludes.

Tentative schedule

Date	*Mandatory Lecture video topics*	*Notes*
Jan-27	• Introduction • Single Processor Machines: Memory Hierarchies and Processor Features	Assign: HW#0
Feb-03	• Parallel Machines and Programming Models • Sources of Parallelism and Locality in Simulation - Part 1	Due: HW#0 Assign: HW#1
Feb-10	• Sources of Parallelism and Locality in Simulation - Part 2 • Shared Memory Programming: Threads and OpenMP, and Tricks with Trees	
Feb-17	• Distributed Memory Machines and Programming • Partitioned Global Address Space Programming with Unified Parallel C (UPC) and UPC++, by Kathy Yelick	Due: HW#1 Assign: HW#2 (Part 1)
Feb-24	• An Introduction to CUDA/OpenCL and Graphics Processors (GPUs), by Forrest Iandola • Dense Linear Algebra (Part 1)	
Mar-03	• Dense Linear Algebra (Part 2): Comm Avoiding Algorithms • Graph Partitioning	Assign: HW #2 (Part 2) Due: HW#2 (Part 1)
Mar-10	• Automatic Performance Tuning and Sparse Matrix Vector Multiplication • Automatic Performance Tuning and Sparse Matrix Vector Multiplication (continued)	
Mar-17	• Structured Grids • Parallel Graph Algorithms, by Aydin Buluc	Due: HW#2 (Part 2) Assign: HW#3
Mar-24	• Fast Fourier Transform	Due: Final Project Proposal
Mar-31	• Dynamic Load Balancing	
Apr-07	• Hierarchical Methods for the N-Body Problem	
Apr-14	• Communication Lower Bounds and Optimal Algorithms	Due: HW#3
Apr-28	**FINAL PROJECT PAPERS DUE**	Due: Final Project Paper
May-05	**PROJECT PRESENTATIONS**	Due: Final Project Presentation