CSCI 338: Parallel Processing

Syllabus for Fall 2019

General Info

Instructor:	Kelly Shaw
Email:	kshaw@cs.williams.edu
Office:	TPL 315
Lectures:	MR 1:10-2:25 TCL 206
Webpage:	http://www.cs.williams.edu/~kshaw/cs338/index.html
Textbooks:	*Unix Systems Programming: Communications, Concurrency, and Threads* (2nd Edition) by Kay Robbins and Steve Robbins *An Introduction to Parallel Programming* by Peter S. Pacheco *Programming Massively Parallel Processors: A Hands-on Approach* (3rd Edition) by David B. Kirk and Wen-mei W. Hwu *Programming on Parallel Machines* by Norm Matloff

Course Description

This course explores different parallel programming paradigms used for writing applications on today's parallel computer systems. The course will introduce concurrency (i.e. multiple simultaneous computations) and the synchronization primitives that allow for the creation of correct concurrent applications. It will examine how a variety of systems organize parallel processing resources and enable users to write parallel programs for these systems. Covered programming paradigms will include multiprogramming with processes, message passing, threading in shared memory multiprocessors, vector processing, graphics processor programming, transactions, MapReduce, and other forms of programming for the cloud. Class discussion is based on assigned readings. Assignments provide students the opportunity to develop proficiency in writing software using different parallel programming paradigms.

Grading Details

Grades will be computed as follows:

5%	Class participation
10%	Reading assignments
35%	Programming assignments
30%	Midterm
20%	Final project

Each of these items are explained in detail in the following sections. **In general, beyond the 3 hours we spend together during our class meeting time, you should expect to spend (on average) at least 10 hours per week on work related to class.**

Class Participation

Lectures are mandatory, and you are expected to attend regularly. One goal of this course is to promote discussion of the assigned topics among all class members. As such, you are encouraged to ask questions, point out problems, and make observations during class.

Reading Assignments

For each assigned reading, students will be asked to answer a small number of questions about the reading before lecture. While discussion of the assigned readings is encouraged, students must complete these questions on their own and should not discuss answers to the questions prior to the due date and time. **Answers for reading assignments will be due before the beginning of the corresponding lecture. Late submissions will not receive credit.** The two lowest scores on reading assignments will be dropped.

Programming Assignments

There will be multiple programming assignments in this course. All programming assignments will be graded based on design, documentation and style, correctness, and efficiency. Programs should be turned in electronically by the posted due date. Each student may use a maximum of **three late days** during the course of the semester. A late day permits you to hand in an assignment up to 24 hours late, without penalty. Once those late days are exhausted, late programming assignments will be penalized 25% per day. Programming assignments will not be accepted more than four days late. To use a late day, you must email the instructor in advance and include a note in the README file accompanying your submission. You may not use late days on the final project or on reading assignments.

We will primarily use the Computer Science Department's Linux computers for our programming assignments. Due to the nature of the course material, different libraries and compilation environments will be used for different assignments, but most assignments will be completed using the C/C++ programming languages. More complex programming assignments will be worth more points towards your final grade than simpler programming assignments.

Midterm Exam

There is a written midterm exam in this course. The exam are closed book and closed notes. It will stress conceptual understanding of the material. Details regarding the specific format of the exam will be discussed in class. **The midterm exam will be Thursday, October 31, in class.**

Final Project

In lieu of a final exam (and in addition to the programming projects described above), you will complete a final project on a topic of your choosing. I will post some sample project ideas on the course webpage later in the semester. I encourage you to be creative and pick a topic related to parallel programming that will be fun and challenging to explore. You may work with a partner

on your final project. In addition to completing the programming aspect of the project, you and your partner will present your results to your classmates and write a paper describing your work. The final project paper and project code will be due during Reading Period/Finals Week (at 5pm on the third-to-last day of the exam period). No extensions are possible as this is a college policy.

Collaboration

Unless specified, programming assignments, reading assignments, and examinations are to be completed individually. I encourage collaboration and assistance in understanding material but not in developing solutions. When solving programming assignments, discussing how to use different libraries (i.e. syntax) and looking up online how to use different libraries are allowed. However, discussing or searching for solutions to questions on reading assignments, programming assignments, and exams is explicitly not allowed. **Please be sure to give explicit credit for any help received.** If you have any doubts about this, ask me whether or not collaboration is appropriate. Uncredited collaborations will be considered a violation of the honor code. The Computer Science honor code and computer usage policy applies to all material in this class. Please review this info at `https://csci.williams.edu/the-cs-honor-code-and-computer-usage-policy/`.

Help

Reaching out for help is an important skill. There are many resources available that I encourage you to consider using. You are encouraged to discuss any questions, concerns, difficulties, or thoughts about the course with the instructor. Other folks on campus want to support you too, including the dean and:

- Math and Science Resource Center: Support is available for students grappling with more quantitative aspects of their coursework. `https://academic-resources.williams.edu/peer-academic-support/math-science`.
- Accessible Education and Disability Support Center: Some students with documented disabilities may require accommodations in certain situations. If this applies to you, please take advantage of the options available. `https:academic-resources.williams.edu/disabilities`.
- The Health Center: Sometimes challenges are not related to courses. The Health Center provides a range of medical, psychological, and health/wellness services. `https://health.williams.edu`.

Inclusivity

The Williams community embraces diversity of age, background, beliefs, ethnicity, gender, gender identity, gender expression, national origin, religious affiliation, sexual orientation, and other visible and non-visible categories. I welcome all students in this course and expect that all students contribute to a respectful, welcoming and inclusive environment. If you have any concerns about classroom climate, please come to me to share your concern.

Topics

The following is a rough outline of topics that we will cover in this class. We will focus more on the highlighted topics.

- **Multiprogramming with processes**
- Message passing
- **Threads and synchronization**
- Transactional memory
- Vector processing
- **Graphics processing**
- Artificial intelligence processing
- **MapReduce**
- Cloud computing