CPEG/ELEG 652: Principles of Parallel Computer Architectures  
Spring 2020  
*Tentative Syllabus*

**Title:**  
Principles of Parallel Computer Architectures

**Time/Room:**  
Tuesdays 3:35pm - 6:35pm,  
Purnell Hall Room 324B  
First class: February 11;  
Last Class: May 12;  
No class on March 31 (Spring Break)

**Important dates:**  
Project proposal due: Friday, February 21 (automatic extension through the weekend)  
Final project report due: Friday, May 15, 23:59 (hard deadline)  
Midterm exam: week of March 23 (schedule individual oral exam two weeks in advance)  
Final exam: finals week (schedule individual oral exam two weeks in advance)

**Instructor:**  
Rudolf Eigenmann  
eigenman@udel.edu  
202 Evans Hall, x0678

**Communication and Office Hours:**  
We will reserve some time at the beginning of each lecture for questions and answers.  
You are welcome to stop by when you see my office door open.  
You can also call or send me email for an appointment.  
I will try to respond to emails within 24hrs. If I haven’t, be sure to let me know in class.

**Course Description:**  
This course provides an introduction to the principles of parallel computer architecture, both hardware and software, with examples of real-world systems and applications. Topcis include, but are not limited to: relevace of parallel computing today; levels of parallelism; programming models with abstract view of the underlying architecture; components of parallel architectures; system topologies; special-purpose architectures; parallel programming languages, libraries, and tools; porting and optimizing parallel applications; performance evaluation; national and international parallel systems; parallel applications.

**What You Will Learn in This Course:**  
All of today’s computer architectures are parallel. Computer architects, developers of the software stack, application programmers, and even end users need to be aware of the way the omnipresent parallelism affects their work. In this course, you will learn the basic structures of parallel hardware and software, how parallel hardware is exposed to system and application software, and how software can best take advantage of the potential offered by parallelism. As time permits, the end-user view will be discussed, including how to access large, national parallel computer systems for computational science.
projects, and how to exploit parallelism if your primary objective is to do physics, chemistry, or biology. This course will expose you to a large range of topics related to parallelism, so it is introductory. For in-depth studies of the materials you will have to take additional, advanced course offerings.

Pre-requisites:
Begin at a level that assumes experience in introductory undergraduate courses such as digital system design, computer architecture, and microprocessor based systems.

Textbooks and other Resources:
These books are recommended resources. They include much more than presented in the course. The lectures will include additional material from research papers and other sources.
• Parallel Computer Organization and Design by Michel Dubois, Murali Annavaram, and Per Stenstrom, Cambridge 2012.
• Using OpenMP, Portable Shared Memory Parallel Programming by Barbara Chapman, Gabriele Jost and Ruud van der Pas, MIT Press 2007.
• Beginning MPI (An Introduction in C) by Wesley Kendall, Amazon Digital Services LLC 2013.
• Parallel Programming in C with MPI and OpenMP by Michael J. Quinn, McGraw-Hill Education 2008.

Online lecture material:
piazza.com/illinois/fall2018/cs420cse402ece492/resources
https://computing.llnl.gov/tutorials/parallel_comp

Grading:
• Midterm: 20%
• Final exam: 30%
• Projects: 25%
• Homework: 15%
• Class Participation: 10%

Approximate schedule:

week
1. Introduction and relevance of parallel computing today. Levels of parallelism
2. Programming models with abstract architectures
3. /
4. Accelerators and special-purpose architectures
5. /
6. Parallel programming languages, libraries, and tools

7. Midterm Exam
   --- Spring Break ---

8. Porting and optimizing parallel applications
9. Porting and optimizing parallel applications (continued)
10. Performance evaluation
11. National and international parallel high-performance computer systems
12. Parallel applications
13. Review

14. Final Exam

Exams:
Both the midterm and the final are individual oral exams of 20 minutes, each. At least two weeks before exam time, please schedule an exam appointment with the instructor
   • in the week of March 23, for the midterm exam, and
   • in the final exam week (optionally in the last week of classes)
The midterm exam will cover material discussed in class up to that point. The final exam is comprehensive.

At the beginning of most lectures we will discuss questions in the style that will be used in exams. Please bring a notepad to class to write down your answers. Exams will be conversation style. A common question is “tell me what you have learned in class”. There will be at least one question about the syllabus.

Homework:
The homework assignment each week will be to do background research on any topic related to the material presented in class that week. In half a page to one page, provide a reference to the material you found (a paper, a textbook, a magazine article, a video, a quote from a person, etc.) and describe in what way the information supports, contradicts, or complements the material presented in class.

Format and submission: Send your homework before the end of each week to eigenman@udel.edu with subject line “CPEG/ELEG652 Homework for week X”.

The expectation is that you will spend one to two hours per week on this assignment.

Projects:
Every student will propose a computational application that you will execute on a parallel machine of your choice. Improve the application in some way that enhances parallelism and measures the resulting performance. Describe your experience in 4-6 pages.

The objective of this project is that you gain experience in
   1. identifying code sections in a program that should/can be parallelized,
   2. learning about a parallel programming model and language,
3. possibly installing a compiler and/or library to facilitate your project,
4. identifying a parallel machine on which to run and measure the program, and
5. analyzing parallel program performance.

For obtaining a good grade, what counts is not so much that you can speed up your program, but that you demonstrate in your report an understanding of these five points. The expectation is that you spend two to three hours per week on this project.

Please create a Google folder in which you place all your project files, including a copy of the project proposal and final project report. In addition, please create a “lab book”, in which you record what you do. Date every entry. There should be at least one entry per week that briefly described what you have done, or say “I was not able to work on the project this week.”

*Format and submission:* At the end of the second week of classes, submit a project proposal of \( \frac{1}{2} \) to 1 page. The proposal should begin with the course number, a project title, and your name. Email the proposal to the instructor with subject line CPEG/ELEG652 Project Proposal.

By the end of the semester, expand this text into a 4-6 page report and submit by Friday May 15 23:59 (hard deadline) with subject line CPEG/ELEG652 Project Report Your Name. The report must include subsections that describe your experience gained w.r.t each of the five project objectives. Be sure that you explain all figures you include sufficiently.

**Policies:**

*Policy regarding class absences and late homework/projects:* If you miss a class, make sure you ask another student in class to inform you of relevant information that may not be captured on the videos. If you have a conflict with an exam date that you know at the beginning of the semester, discuss alternative ways of taking the exam with the instructor within the first two weeks of classes. Conflicts that come up later, you will need to resolve with the party that created them. Exceptions to this policy require students to work with Assistant Dean Shermeyer to submit proof of extenuating circumstances so that the Dean's office emails the instructor an official Excused Absence Notification (EAN). If you receive an EAN, you need to then contact the instructor to discuss alternatives to taking the exam.

Be sure you do homework and projects early, allowing for possible emergencies (such as illness or computer failed just before project submission).

UD has a number of policies to make learning and campus life as smooth and enjoyable as possible for everyone. Please familiarize yourself with these policies.

[http://www1.udel.edu/stuguide/19-20/code.html](http://www1.udel.edu/stuguide/19-20/code.html)