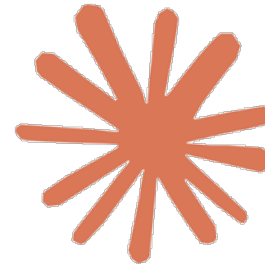


18-647

Computational Problem Solving for Engineers

Course Mechanics

Franz Franchetti
Instructor



Quentin Oschatz
Teaching Assistant



Chenxi Wan
Teaching Assistant



This Course in a Nutshell

Step 1:

$$75 \div 4 = 18 \text{ R } 3$$

divisor 18 quotient

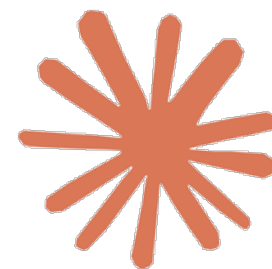
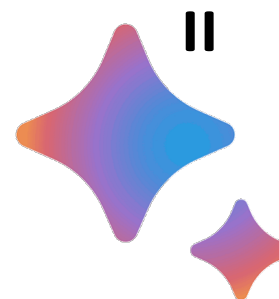
$$\begin{array}{r} 4 \overline{) 75} \\ - 4 \\ \hline 35 \\ - 32 \\ \hline 3 \end{array}$$

dividend

remainder 3

||
18-647

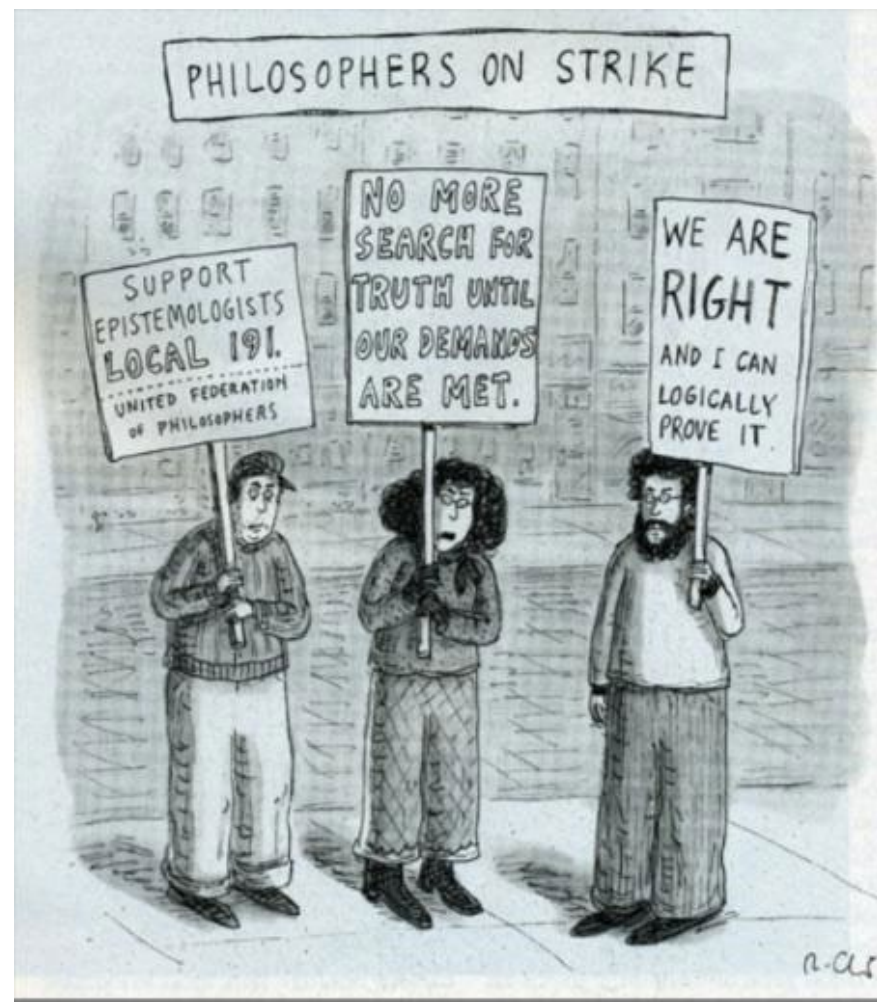
Step 2:



*We all learned arithmetic before getting a pocket calculator.
Same holds for Generative AI.*

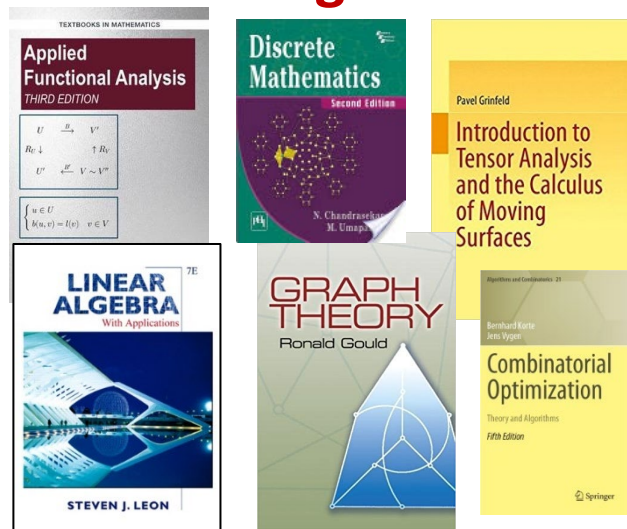
Course Philosophy

- Learn how to solve real engineering problems with large(r) computers
- Learn how to use AI when solving engineering problems
- **Your productivity is key: shortest time to solution**
- Learn how to scale from laptop to large machines and the cloud productively
- No need to become a performance engineer, but *know your math and computer stuff*



18-647: 10,000 Foot View

Scalable Algorithms



Larger-Scale Hardware



Languages and Libraries

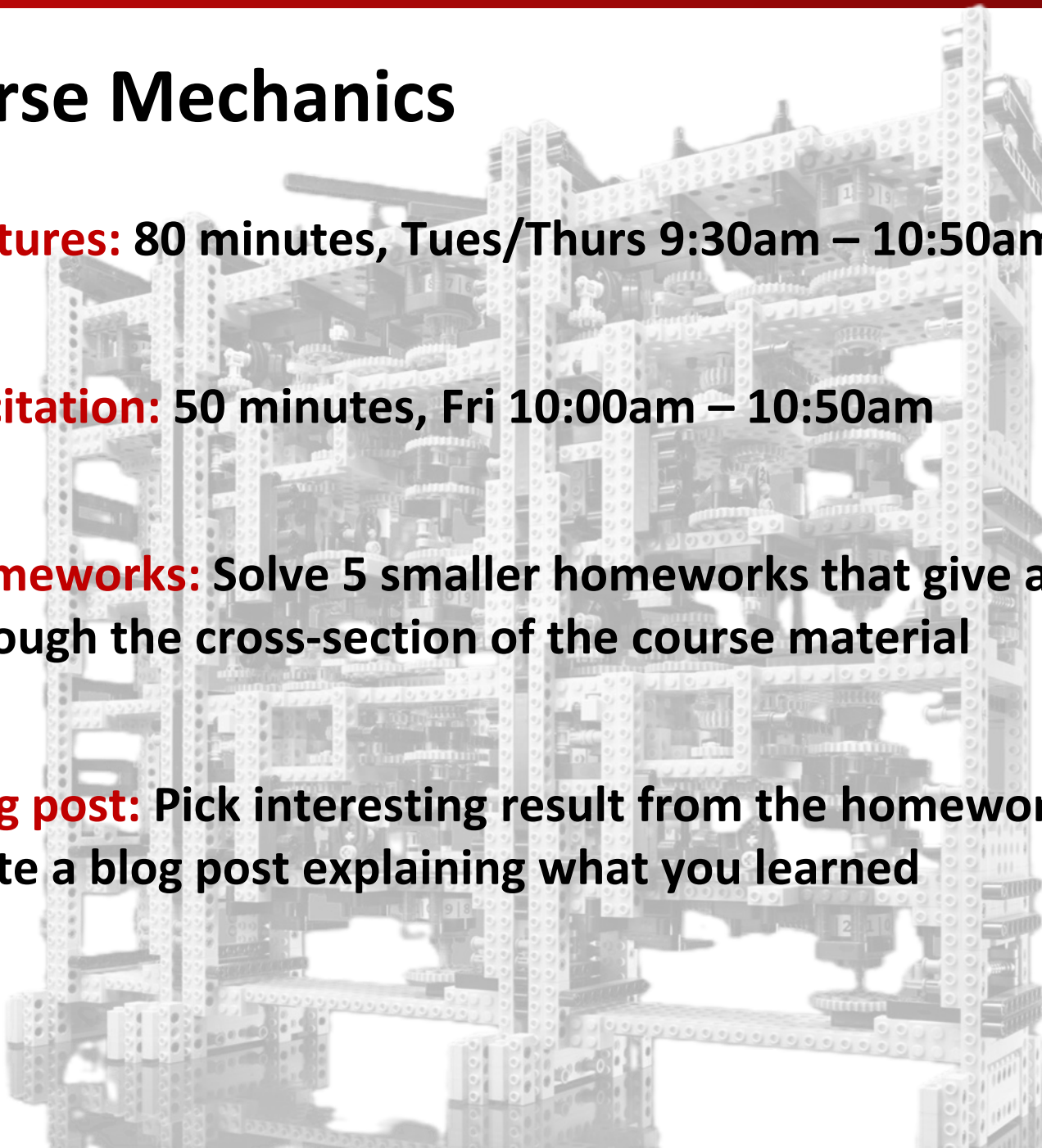


Performance vs. Productivity



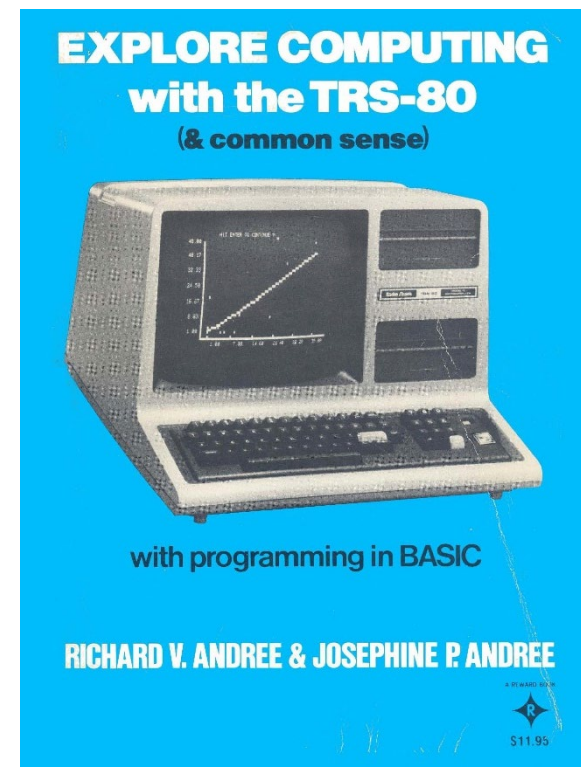
Course Mechanics

- **Lectures:** 80 minutes, Tues/Thurs 9:30am – 10:50am
- **Recitation:** 50 minutes, Fri 10:00am – 10:50am
- **Homeworks:** Solve 5 smaller homeworks that give a tour through the cross-section of the course material
- **Blog post:** Pick interesting result from the homeworks and write a blog post explaining what you learned



5 Homeworks: Explore Computing

- **Deeper-dive into algorithm families**
Algorithms underlying CSE, AI and ML
- **Explore performance vs. productivity**
High level/scripting vs. high performance languages
- **Explore range of computer systems**
Linux servers, cloud, HPC, high throughput
- **High productivity through specialization**
Frameworks and problem solving environments
- **Assess how good is your solution**
Profiling, timing, modeling, assessing results



Homework Nuts And Bolts

- **Pre-read: We will give a short paragraph out**

Do your research to be prepared
AI use is OK, no coding

- **Plan and discuss: in recitation**

In small groups plan out how to solve the problem, discuss, present

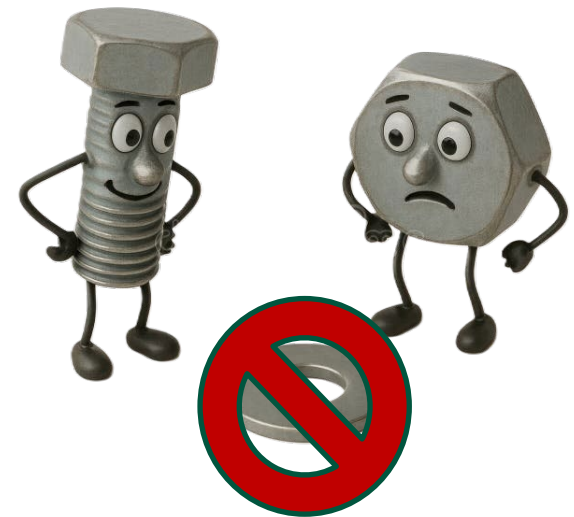
- **Coding: actually solve the problem**

OK to use any approved AI tool,
document AI usage

- **Post-mortem: In-class discussion**

Instructor and Tas explain solution and background
Student discussion of lessons learned and surprises

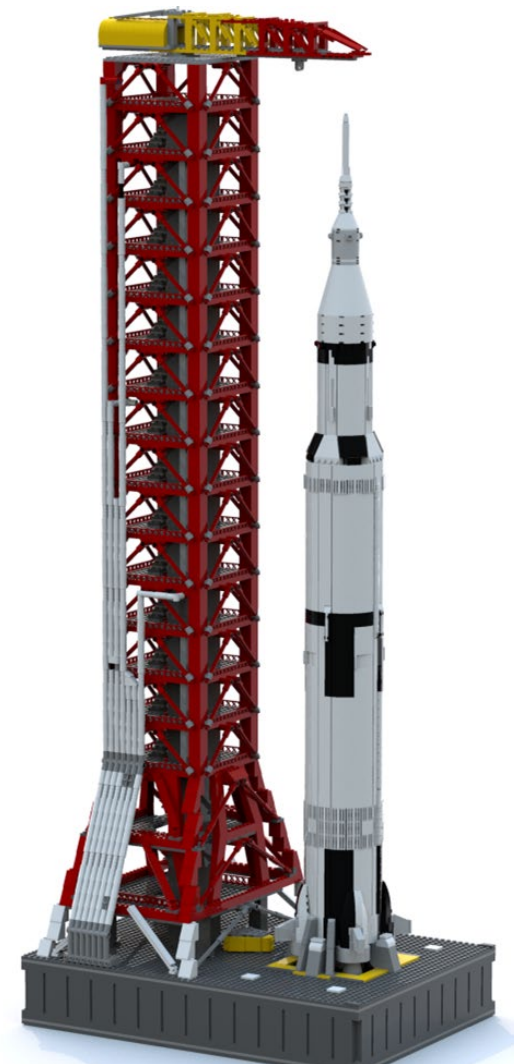
Any free CMU-approved AI tool can be used. No personal tokens



Blog Post: For An Interesting HW Problem...

- Problem specification
- Algorithm
- Hardware and software
- Outcome
- Scalability and complexity
- Quality of the result
- Performance analysis

AI Usage: Ok within the rules

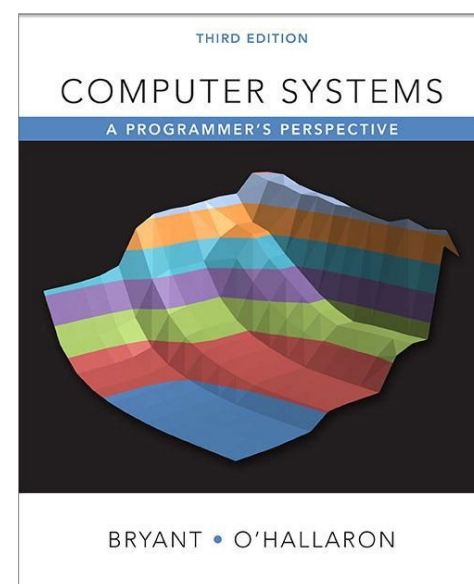
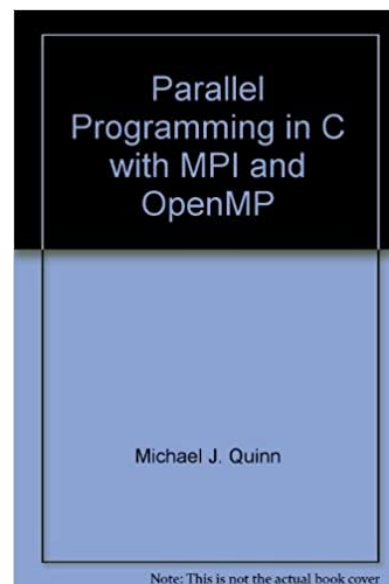
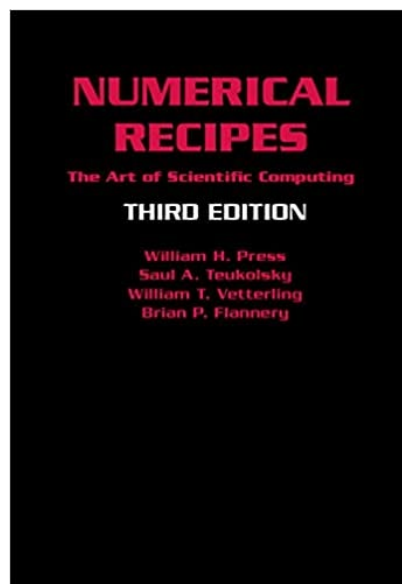
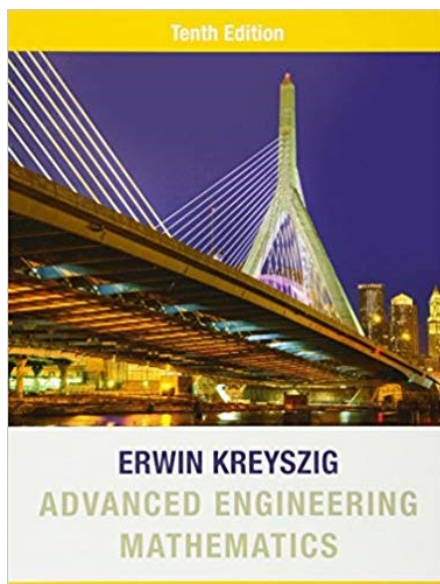


Grading

- In-class participation: 30%
(lecture and recitations)
- Blog post: 20%
- 5 Homeworks: 50%



18-647 Additional Material



Advanced Engineering Mathematics, 10th Edition

Edward J. Norminton, Herbert Kreyszig, Erwin Kreyszig

Computer Systems: A Programmer's Perspective, Third Edition

Randal E. Bryant and David R. O'Hallaron, Carnegie Mellon University

Numerical Recipes: The Art of Scientific Computing

William H. Press, Saul A. Teukolsky, William T. Vetterling and Brian P. Flannery

Parallel Programming In C with MPI and OpenMP

Michael Quinn

There is no course text book but these books cover the space

Tentative Schedule (Guest Lectures may move)

Dates	Topic
August 25	State of computing: What is the current state of the art from embedded devices through desktops, servers, and consumer system all the way to cloud, HPC, and supercomputing
August 27	Computer architecture: Relevant computer architecture concepts
September 1	Software stack: ISA, operating system, virtualization, messaging, containers
September 3	Mathematics for Engineers: The central role of numerical linear algebra
September 8	Algorithm analysis, scalability, complexity: Getting answers in time
September 10	The ECE Computing Environment: number cluster, Andrew systems, capability machines, GPU access, cloud access, Pittsburgh Supercomputing Center and XSEDE
September 15	Parallelization: Sequential vs. parallel algorithms, scalability vs. performance
September 17	Need for speed: Principles of code optimization, when and how to optimize code
September 22	Cloud computing and HPC: Amazon EC2/Windows Azure/Google Cloud, Computational Grids, Scientific Workflows, Computing Centers
September 24	Numerical Analysis: How good are your answers? How to make them better?
September 29	Scalable algorithms: Dense numerical linear algebra, CNNs/DNNs, FFTs
October 1	Scalable algorithms: Graph algorithms and sparse numerical linear algebra

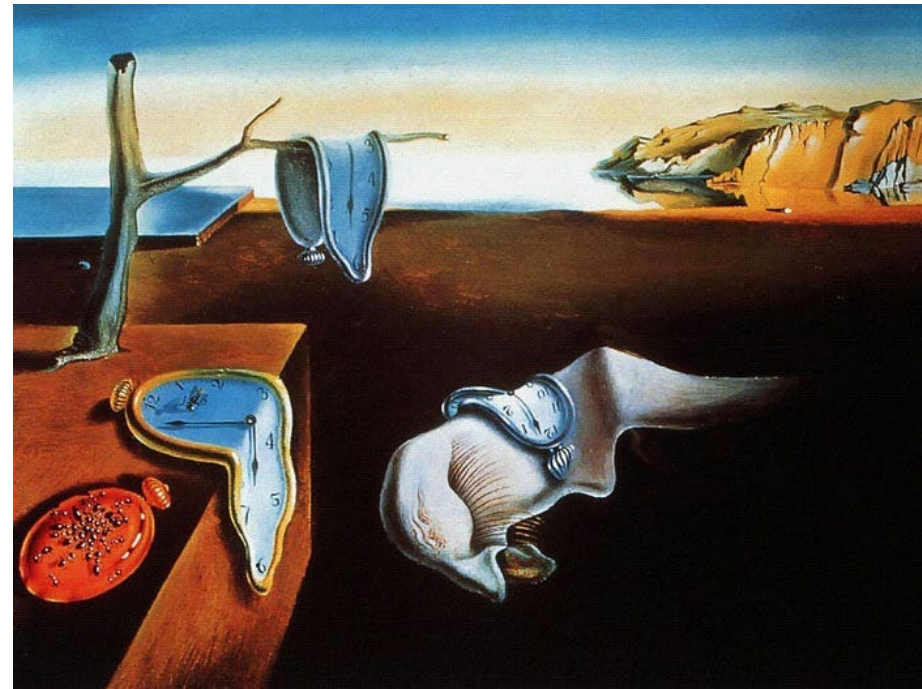
Schedule (Cont'd)

Dates	Topic
October 6	Scalable algorithms: ODE and PDE solvers, stencils, filters, discretization
October 8	Scalable algorithms: Discrete and continuous optimization, ML training
October 20	Scalable algorithms: Informatics, symbolic computing, higher level AI
October 22	Scalable algorithms: Statistics: Monte Carlo, MCMC, statistical ML
October 27	Data: Obtaining real data sets, data visualization, data bases, data stores, file systems
October 29	Modern CPUs I: Super-Scalar Out-of-order, multicore
November 5	Guest Lecture: Pittsburgh Supercomputing Center and Usable HPC and High Performance AI/ML
November 10	Modern CPUs II: vector extensions
November 12	Hardware Acceleration: GPUs, FPGAs, TPU, Tensor Cores,...
November 17	Guest Lecture: Quantum Computing and Quantum Algorithms
November 19	From productivity to performance: C++, OpenMP, MPI, CUDA, Autotuning
November 24	Research Talk: SPIRAL: Formal Software Synthesis of Computational Kernels
December 1	Guest Lecture: Software Engineering vs. Performance Engineering
December 3	Summary and Beyond

Homework and Blog Post Schedule

Dates	Topic
September 3	Release homework 1
September 15	Homework 1 due, release homework 2
October 8	Homework 2 due, release homework 3
October 29	Homework 3 due, release homework 4
November 19	Homework 4 due, release homework 5
December 1	Homework 5 due
December 3	Blog post due

***No late days.
But deadlines can be extended***



Teaching Staff



Instructor

Franz Franchetti

Office Hours: Mon 1pm—2pm

Scott Hall 6164

<http://www.ece.cmu.edu/~franzf>

Canvas:

<https://canvas.cmu.edu/courses/54903>



Teaching Assistant

Quentin Oschatz

Office Hours: TBD



Teaching Assistant

Chenxi Wan

Office Hours: TBD

TAs, Recitation, and Office Hours

- TAs are mentors for your blog post and give advise regarding homeworks and system issues
- Weekly recitation to help with homework issues that may be prevalent across entire class
- Weekly faculty office hours to discuss homeworks and general course related topics
- Faculty and TA one-on-ones if needed, schedule via email



Other Logistics

- **ECE IT Helpdesk**
email: help@ece.cmu.edu
- **ECE Course Support: HH 1113**
email: ece-asc@andrew.cmu.edu
- **Accounts: ACCESS, ECE, Amazon EC2,...**
- **Software access/licenses**
- **Approved AI:**
<https://www.cmu.edu/computing/services/ai/tools/index.html>
- **ECE ITS and Pittsburgh Supercomputing Center (PSC)**
- **Orchard helpdesk**

Enjoy

