Introduction to Computer Systems

Syllabus

Class Resources

Course Website	http://pdinda.org/ics All public course details, syllabus, schedule, etc.
~cs213/HANDOUT	Directory on course server Some private materials
Canvas	https://canvas.northwestern.edu/courses/237534 Grade reports, occasional zoom stuff, Gradescope, link to class calendar
Piazza	https://piazza.com/class/mf8xw04hcc16n2 Discussion. we will enroll you.
Gradescope	Accessed via Canvas. Some handins. We will enroll you.

This course follows the Northwestern University Syllabus Standards, which can be found at https://www.registrar.northwestern.edu/registration-graduation/northwestern-university-syllabus-standards.html Students are responsible for familiarizing themselves with this information.

Office hours and discussion/recitation times are available via a shared calendar linked from the course web page and on Canvas. Our goal is to have every student be able to attend at least one office hour per week, and for the maximum possible number of students be able to attend the optional discussion/recitation.

Class size and content ultimately depends on TA support.

This syllabus is subject to change. We will notify students of changes in lecture and/or on Piazza.

Instructor

Peter A. Dinda
Mudd 3507
pdinda@northwestern.edu

Teaching Assistants and Peer Mentors

Alex Butler (TA)
Mudd 3301
lxbtlr@u.northwestern.edu

Harry Guan (PM)
harryguan2027@u.northwestern.edu

Christian Lee (PM)
christianlee2027@u.northwestern.edu

Ryan Murphy (PM)
ryanmurphy2026@u.northwestern.edu

Ethan Sterling (PM)
ethansterling2026@u.northwestern.edu

Haresh Wedanayake (PM)
hareshwedanayake2027@u.northwestern.edu

Leo Zhang (PM)
leozhang2027@u.northwestern.edu

Location and Time

Lectures:	Tuesdays and Thursdays, 9:30-10:50, Frances Searle 1421
Midterm Exam:	TBD, mid-quarter, outside of class, in person, on paper
Final Exam:	(officially) Monday, 12/8, 12pm, in person, on paper

Prerequisites

Required	CS 211 or equivalent
Required	Experience with C or C++
Required	Some experience with programming in a Unix environment (e.g., as in CS 211)

CS 213 is a **required core course** in the Computer Science curriculum in both McCormick and Weinberg. It is also a required course for CS minors in both schools. 213 can also be taken for credit within the Computer Engineering curriculum. 300-level systems courses generally have 213 as a prerequisite.

Textbook

Randal E. Bryant and David R. O'Hallaron, *Computer Systems: A Programmer's Perspective, **Third Edition**,* Prentice Hall, 2015, (ISBN-13: 978-0134092669, ISBN-10**:** 013409266X) (Required - Textbook)

- Details on http://csapp.cs.cmu.edu
- **Make sure you have the third edition of the book**. This edition is the first to focus on the 64 bit operation of the machine, which we will make extensive use of in this course.
- If you buy a non-U.S. version, acquire a pdf through some means, etc, please be aware that these can have differences from the U.S. version. In particular, for any homework question assigned from the book, please be sure to use a U.S. version. The U.S. version should be available in the library.
- There is now an electronic version of this book available for rent. For details, see the CS:APP web site.

Brian W. Kernighan and Dennis M. Ritchie, *The C Programming Language, Second Edition*, Prentice Hall, 1988 (ISBN 0-131-10370-9) (Reference)

- This remains the definitive book on C by its creators

Richard Stevens and Stephen Rago, *Advanced Programming in the Unix Environment*, *Third Edition*, Addison-Wesley, 2013 (ISBN-10: 0321637739 | ISBN-13: 978-0321637734) (Reference)

- This describes how to think like a Unix systems programmer
- The older editions, even the first edition, are very good

Objectives, framework, philosophy, and caveats

This course has four purposes. First, you will learn about the hierarchy of abstractions and implementations that comprise a modern computer system. This will provide a conceptual framework that you can then flesh out with courses such as compilers, operating systems, networks, and others. The second purpose is to demystify the machine and the tools that we use to program it. This includes telling you the little details that students usually have to learn by osmosis. In combination, these two purposes will give you the background to understand many different computer systems. The third purpose is to bring you up to speed in doing systems programming in a low-level language in the Unix environment. The final purpose is to prepare you for upper-level courses in systems.

This is a learn-by-doing kind of class. You will write pieces of code, compile them, debug them, disassemble them, measure their performance, optimize them, etc.

The specific computer architecture we will focus on in this class is the 64 bit Intel/AMD x86 architecture, which is used in virtually all supercomputers, clouds, clusters, servers, desktops, and laptop/notebook computers today.[1] The specific

[1] The 64 bit x86 architecture is also called "x86_64" and just "x64". We may also look briefly at the ARM architecture used in iPhones/iPads and many Android devices, and/or the up-and-coming RISC-V

operating system we will use is Linux, which is used in most supercomputer, cloud, cluster, and server environments, and is the operating system of Android smartphones and ChromeBooks. The specific programming toolchain we will use is GCC (and GDB), which is an extremely widely used core toolchain on pretty much all platforms, except Windows. The ideas and concepts embodied in this architecture, operating system, and programming toolchain are commonly found in others.

This course is ideally taken after 211 early in your academic career.

Reading / Lectures / Attendance Requirement

Lectures will be held in person in the scheduled room at the scheduled time. It is important that you complete the reading assigned for each officially scheduled class session before that session (the reading for the first session is an exception). **Based on your reading, you should prepare at least one question for each lecture. You will submit this question on a form in Canvas. This is due by 8AM on the day of the lecture.**

You are required to attend lecture. We use some materials and structure that are different from other instances of 213, and we do not use slides in lecture. If you would also like to see slides, you can find a pointer to CMU slides on our course web site. These slides mirror the book closely. Branden Ghena here at Northwestern also teaches using slides and makes them publicly available.

You are encouraged to ask questions or provide comments, either verbally, in chat, in office hours, or on Piazza. You can also help answer other student's questions and comments. What I'm asking of you is: Read. Attend. Ask. Answer. There is no such thing as a dumb question (or too esoteric of a question) - we will try our best to answer or comment on all questions.

Getting Help

Your instructor, TA, and PMs have regularly scheduled office hours and are available by appointment if these do not work. We have scheduled our office hours to try to spread them across the week. The course calendar is linked to from the Canvas home page.

We will use an online discussion group on Piazza as well. We will enroll you. The link is on the course web page and on Canvas. The intent is to have multiple venues for discussion with different styles so that all students feel comfortable participating. If you have a question, answer, or comment, please put it forward.

---

architecture. ARM is also the architecture used in "Apple Silicon" machines. RISC-V is an open, public architecture. If this doesn't make sense to you yet, don't worry about it.

If you're too scared, you can ask anonymously to your classmates on Piazza. We will try our best to answer.

Resources

Machines at Northwestern will be available for remote login. You will have access to several server machines that can support many users simultaneously, and we expect most students will use those servers. We will test your labs on those machines. You should also be able to work on labs on your own machine provided it is running a reasonably recent Linux.[2]

For students who find the topics of this course particularly compelling, we can give you access to even more interesting machines.

Labs

We will have four programming labs. Their goal is to make you apply the concepts you've learned and to gain familiarity with Unix tools that can help you apply them. Labs should be done in groups of two. **Start looking for a partner on day one.**

Homeworks

We will give you several graded homework assignments, give you some time to work on them and hand-in the results, and then provide solutions. Homeworks are to be done individually, and are important for preparing for exams. The precise grading criteria will be given at a later time.

Exams

There will be a midterm exam and a final exam. The final exam will not be cumulative. I do not provide practice exams. Instead, we will schedule midterm and final exam review sessions. Exams are not returned.

Grading

50 % Programming labs (4 labs, 12.5% each)
10 % Homeworks (4 assignments, 2.5% each)
20 % Midterm (covers first half of the course, **but see below**)
20 % Final (covers second half of the course, **but see below**)

For some of the programming labs, extra credit is possible.

[2] If you would like to do this, but your machine uses Windows or MacOS (Intel Only), you can install virtualization software, and then install Linux in a virtual machine. We typically use VMware for this (Workstation on a Windows box, Fusion Pro on an (Intel) Mac), but there are other tools. Ubuntu is a reasonably good choice of Linux, although the CS department's servers run Red Hat.

Reading and preparation for lecture is expected. As noted above, you will fill out a web form about your reading by 8AM on the day of the lecture. The TA and PMs will review these submissions and I will attempt to weave the most common questions into lecture. **You may miss up to 5 of these submissions. Missing more will result in you failing the course.**

Your score in the course is the weighted average of your scores on each of the components. You can view all currently graded material, and your score, at any time on Canvas. Final grades are based on the course score (the weighted average), with the basic model being that the 90s are A territory, 80s are B territory, and so on. **HOWEVER, you need to have a passing grade (64% or higher) averaged between the two exams in order to pass the course, regardless of your performance in the homework and lab assignments.** We will strictly follow this rule – there will be no exceptions. We believe we have structured the course reading, lectures, homework, labs, and other opportunities so that it should be reasonably easy to pass this threshold.

The instructor ultimately assigns scores and grades in consultation with the TA and PMs. If you have a problem with a score on an assignment/exam or your grade, you are welcome to bring it up with them or the instructors, but only the instructors are empowered to change grades.

Late Policy

For each calendar day after the due date for a lab, 10% is lost. After 1 day, the maximum score is 90%, after 2 days, 80%, etc, for a maximum of 10 days.

Cheating, AI, and Inadvertent Disclosures

Since cheaters are mostly hurting themselves, we do not have the time or energy to hunt them down. We much prefer that you act collegially and help each other to learn the material and to solve problems than to have you live in fear of our wrath and not talk to each other. Nonetheless, if we detect blatant cheating, we will deal with the cheaters as per Northwestern guidelines. Please note that you are subject to the Academic Integrity Policy.

"Prompt engineering" or "vibe coding" your way through labs and homeworks will make it unlikely that you learn anything. The labs and homeworks are a substantial portion of the grade for a good reason.

If you decide to place your work in a public repository (github, say), be sure that you mark the permissions so that only you and your teammates can access it.

Please do not place class materials on any public site. If it's on the course web site, it's already public and will remain public. If it's from the discussion group or from the handout directory on the course servers, it should not be shared publicly.

Accessibility / ANU

Any student requesting accommodations related to a disability or other condition is required to register with AccessibleNU (accessiblenu@northwestern.edu; 847-467-5530) and provide professors with an accommodation notification from AccessibleNU, preferably within the first two weeks of class. All information will remain confidential.

Schedule

Lecture	Date	Topics	Readings	Homework/Lab
1	9/16 T	Mechanics, Introduction, overview of abstractions	Chapter 1	Pack lab out
2	9/18 Th	Physics, transistors, photolithography, Moore's Law, bits, bytes, logic, cores, and multicores	2, 2.1, physics-to-logic handout	HW1 out
9/22 is the last day for adding courses or changing sections				
3	9/23 T	Integers and integer math	2.2-2.3	
4	9/25 Th	Floating point and FP math	2.4-2.5	
5	9/30 T	The Machine Model - instruction set architecture, microarchitecture, and basic data flow with registers and memory	3, 3.1-3.5, 5.7	HW 1 in, HW 2 out
6	10/2 Th	Control flow	3.6	Pack lab in Bomb lab out
7	10/7 T	Procedures	3.7	
8	10/9 Th	Data	3.8-3.10	
9	10/14 T	Floating point	3.11-3.12	HW 2 in HW 3 out
10	10/16 Th	Slack		
Midterm Exam Review: Around here, time+location TBA – outside of class				
***Midterm Exam**: Around here, time+location TBA – Midterm is outside of class, in person, and on paper*				
11	10/21 T	Memory and cache	6, 6.1-6.4	

12	10/23 Th	Cache performance	6.5-6.7	Bomb lab in, Attack lab out
10/24 is the last day to drop a class				
13	10/28 T	Cache performance / catchup	6.5-6.7	
14	10/30 Th	Linking 	Chapter 7	
15	11/4 T	Concurrency and Parallelism	Chapter 12 (focus on 12.3+), Concurr-ency and Parallelism handouts	
16	11/6 Th	Exceptional control flow	8,8.1-8.4	Attack lab in, SETI lab out
17	11/11 T	Exceptional control flow	8.5-8.8 Unix Nutshell handout	HW 3 in, HW 4 out
18	11/13 Th	Virtual memory Memory system	9, 9.1-9.8	
19	11/18 T	Memory allocation	9.9-9.12	
20	11/20 Th	Input and Output	Chapter 10	
 *Thanksgiving break*				
21	12/2 T	Network Programming	Chapter 11 Sockets Handout	SETI lab in
22	12/4 Th	Special Topic or Slack		HW4 in
Final Exam Review: Around here, time+location TBA – outside of class *Finals week – Exam is officially on Monday, 12/8, 12pm – Exam is in person and on paper. It is not cumulative.*				

Note that in the latter part of the course, we will cover Chapters 10-11 at a very high level. I want you to read these chapters, but I will not cover them in their entirety in class.

We will skip Chapter 4 (Processor Architecture), 5 (Performance Optimization), and others. Chapter 4 is worth reading if you're interested in how a simple processor with an Intel-like instruction set is implemented. Chapter 5 is all about understanding how to make programs run faster.