Course Intro

Instructors: Connor (Cece) McMahon, Jenny Song, Jonathan Shi
Agenda

• Staff Introduction
• Course Overview
• Course Information
Introducing Connor (she/her)

- **Upbringing:** Born in the New Orleans, lived in Atlanta since 8
- **Education:** BS in Computer Engineering from Georgia Tech, Pursing MS in CS at UC Berkeley
- **Teaching:** 61C TA
- **Interests:** Hiking, Traveling
Introducing Jonathan (he/him)

few known pictures of me facing the camera exist

- **Upbringing:** born in the East Bay, raised in the South Bay
- **Education:** finished undergrad this year (EECS), coming back for MS
- **Teaching:** 61C + 162 course staff, formerly 16A CSM
- **Interests:** [dinosaur comics](#), amateur saxophonist, amateur chess player, amateur smash bros. player
Introducing Jenny (she/her)

• **Upbringing:** Born in Hangzhou, China, High school in New Jersey
• **Education:** Berkeley grads, Masters at CMU
• **Teaching:** 61CCC
• **Interests:** Baking, Running, Tiktoking, Procreating
Introducing Your TAs

Anjan Das
Abel Yagubyan
Arunan Thiviyanathan
Aadith Srinivasan
Raghav Singh
Raghav Gupta
Justin Yokota
Jerry Xu
Caroline Liu
Introducing Your Tutors

Amit Narang
Carolyn Duan
Cindy Lin
Edwin Lim
Ella Schwarz
Rosalie Fang
Vinay Guatam
Yuanhan Li
Kenneth Lien
Riley Dyer
Agenda

• Staff Introduction
• Course Overview
• Course Information
1. How do computer processors and memories work, and how do they affect software design and performance?

2. Introduction to “computer systems” areas: architecture, compilers, security, embedded, operating systems, digital design, and more!

→ (CS 152, CS 164, CS 161, CS 149, CS 162, EECS 151, etc.)
Hardware-Software Interface

**Software**

- **Parallel Requests**
  Assigned to computer
  e.g. search “Steven Ho”

- **Parallel Threads**
  Assigned to core
  e.g. lookup, ads

- **Parallel Instructions**
  > 1 instruction @ one time
  e.g. 5 pipelined instructions

- **Parallel Data**
  > 1 data item @ one time
  e.g. add of 4 pairs of words

- **Hardware descriptions**
  All gates functioning in parallel at same time

**Hardware**

- Warehouse Scale Computer
- **Leverage**
  Parallelism & Achieve High Performance
- **Smart Phone**
- **Computer**
  - Core
  - …
  - Core
  - Memory
  - Input/Output
- **Instruction Unit(s)**
  - $A_0 + B_0$
  - $A_1 + B_1$
  - $A_2 + B_2$
  - $A_3 + B_3$

**Logic Gates**
Know the tools of the trade – computers!
  ○ “Computers” come in all shapes and sizes
  ○ Computing achieved in many different ways nowadays

Know how to improve program performance
  ○ Parallelism techniques
  ○ OS + computer architecture basics

Design large systems – abstraction in hardware

Design methodology – limitations and tradeoffs
Course Learning Objectives

After taking this class students should be able to:

✓ Identify and explain the various layers of abstraction that allow computer users to perform complex software tasks without understanding what the computer hardware is actually doing

✓ Judge the effect of changing computer components (e.g. processor, RAM, HDD, cache) on the performance of a computer program

✓ Understand how the memory hierarchy enables fast memory accesses

✓ Construct a working CPU from logic gates for a specified instruction set architecture

✓ Identify the different types of parallelism and predict their effects on different types of applications
Course Learning Objectives

In addition, this class will require students to work on the following skills:

– Creating and modifying designs to meet a given set of specifications

– Identifying unexpected or problematic situations using debugging tools, and creating test cases to ensure proper behavior

– Defending design choices based on tradeoffs and limitations
Six Great Ideas in Computer Architecture

1) Abstraction
2) Moore’s Law
3) Principle of Locality/Memory Hierarchy
4) Parallelism
5) Performance Measurement & Improvement
6) Dependability via Redundancy
Agenda

• Staff Introduction
• Course Overview
• Course Information
Course Information

• **Course Website:** [https://cs61c.org/](https://cs61c.org/)
  — Check for weekly schedule, assignments, staff contact info

• **Ed:**
  — Core platform for announcement, discussion and clarification

• **PrairieLearn**
  — Home assignments and exam

• **Gradescope**
  — Lab and project

• **Inst account:** [http://inst.eecs.berkeley.edu/webacct](http://inst.eecs.berkeley.edu/webacct)
  — Access instructional (“hive”) machines for project and lab work

• **Textbook**
# Course elements

<table>
<thead>
<tr>
<th>Lecture</th>
<th>Lecture content is pre-recorded all delivered asynchronously.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lab</td>
<td>We encourage you to work with a partner (in your own zoom rooms) on your own time. Labs are auto-graded on Gradescope based on correctness.</td>
</tr>
<tr>
<td>Discussion</td>
<td>3 Live 1hr discussions and a 2hrs lost section are host on W/F. Pre-recorded discussion of TAs going over the worksheet are also available.</td>
</tr>
<tr>
<td>Office Hour</td>
<td>OHs are held on zoom throughout the day. We have dedicated project OH, and regular OH.</td>
</tr>
<tr>
<td>Project</td>
<td>4 projects graded on correctness and ran with plagiarism detection programs.</td>
</tr>
<tr>
<td>Homework</td>
<td>Weekly homework, you will have infinite tries, graded on correctness.</td>
</tr>
<tr>
<td>Exam</td>
<td>1 midterm and a final, both proctored, alternative time request will be available.</td>
</tr>
</tbody>
</table>
Course Assignments and Grading

• **Labs** (10%) – 30 pts

• **Homework** (11%) – 33 pts

• **Projects** (48%) – 4 total, weighted equally
  - Proj1 (12%) – Proj2 (12%)
  - Proj3 (12%) – Proj4 (12%)

• **Exams** (31%)
  - **Midterm** (11%): Thursday 7/22 9:30AM
  - **Final** (20%): Thursday 8/12 9:30AM

• **EPA** *(Effort Participation Altruism)*
  - Attendance and active engagement in course events
  - (eg: discussions, OHs, review sessions, guerilla sessions, tutoring, etc)
Project Partners

• You may work with a partner for projects

• We will have an Ed thread for you to find partners, also a meet-n-greet session, partner-matching, more on ed post

• Pair programming works great!

• Guidelines: https://cs61a.org/articles/pair-programming/
Exam Clobbering & Conflicts

- Midterm is clobber-able with your score on the final exam!
- We will offer a full (z-score) clobber to the midterm from the final. This means that your midterm score will be the max of what your current score and the z-score of the final mapped to the midterm.
Late Policy – Slip Days

- Assignment submissions due at 11:59pm
- **Homework**: 3 slip days, 1 drop, no late credit after slip days
- **Lab**: 1 drop, half credit if submitted within a week late, no credit after that
- **Project**: 3 slip days, 33% deduction of score per day after slip days

- Slip day tokens will be distributed amongst your late submissions at the end of the semester to maximize your grade to benefit to having leftover tokens.

- Use the slip tokens at own risk – don’t want to fall too far behind.
- If you have an emergency, request an extension!
EECS Grading Policy

- [http://www.eecs.berkeley.edu/Policies/ugrad.grading.shtml](http://www.eecs.berkeley.edu/Policies/ugrad.grading.shtml)

“A typical GPA for a lower division course will fall in the range 2.8 - 3.3, depending on the course and the students who enroll. For example, a GPA of 3.0 would result from 35% A's, 45% B's, 13% C's, and 7% D's and F's.”
61C Grade Bins

- Course is graded out of 300 points
- If the grade bins result in an average GPA that is too low, the course will be curved to match department guidelines BEFORE adding EPA.

<table>
<thead>
<tr>
<th>Raw Score</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>290+</td>
<td>A+</td>
</tr>
<tr>
<td>[270,290)</td>
<td>A</td>
</tr>
<tr>
<td>[260,270)</td>
<td>A-</td>
</tr>
<tr>
<td>[250,260)</td>
<td>B+</td>
</tr>
<tr>
<td>[230,250)</td>
<td>B</td>
</tr>
<tr>
<td>[220,230)</td>
<td>B-</td>
</tr>
<tr>
<td>[210,220)</td>
<td>C+</td>
</tr>
<tr>
<td>[190,210)</td>
<td>C</td>
</tr>
<tr>
<td>[180,190)</td>
<td>C-</td>
</tr>
<tr>
<td>[140,180)</td>
<td>D</td>
</tr>
<tr>
<td>[0,140)</td>
<td>F</td>
</tr>
</tbody>
</table>
Policy on Assignments and Independent Work

• We understand that this class is stressful and can become overwhelming
• You learn by doing, not cheating
• When students cheat, it is impossible for instructors to understand what material the students are struggling with
• Cheating hurts you, your classmates, and your professors
• If you need help, please reach out to us
Policy on Assignments and Independent Work

• All submissions should be completed by you or you and your partner alone
• You are encouraged to discuss your assignments with other students (ideas), but we expect that what you turn in is yours
• It is **NOT acceptable** to copy solutions from other students
• It is **NOT acceptable** to copy (or start your) solutions from the Web (including Github)
• More information on the course policy webpage
Policy on Assignments and Independent Work

• We have tools and methods, developed over many years, for detecting cheating. You WILL be caught, and the penalties WILL be severe.

• Both the cheater and the enabler receive $-100\%$ for the assignment. Letter to your university record documenting the incidence of cheating.

  — IT IS BETTER TO NOT DO THE ASSIGNMENT THEN TO CHEAT

• People are caught every semester of 61C
Tutor Resources

- Weekly Guerrilla Sessions (synchronous but recorded)
  - Review sessions covering exam problems on specific topics
- Small-group tutoring (synchronous NOT recorded)
  - Sign up for a weekly tutoring session w/ a tutor
  - Groups are as small as 5-10 people; personalized help
  - Sign-ups will be posted on Ed
Successful Behaviors

● Practice, practice, practice
  ○ Learn by doing: deep learning doesn’t happen in lecture (and it shouldn’t!)
  ○ Growth mindset: success through effort and repetition

● Find a learning community
  ○ Learning is much more fun with friends
  ○ Learn via discussion of concepts with other students

● Avoid comparison
  ○ do your best, and judge yourself on your progress alone.
  ○ Remember, we have the clobber policy!!

● You learn best from your mistakes
  ○ Don’t be afraid to be wrong; you are here to learn, please ask us any questions you may have
This Week

• Intro form
• Lab 0 released today, due Friday
• Lab 1 released Thursday, due next Monday
• Homework 1 released this afternoon (will make Ed post), due next Monday
• Homework 2 released Wednesday, due next Tuesday
• Project 1 released Friday, due next Friday
Reminder

• Please only email us for private matters. All other questions should be posted on Ed.
Thank you and good luck!