Lecture 1: Course overview

Brennon Brimhall

2 June 2025

Welcome!

- Welcome to CSF!
- Today:
  - Administrative stuff
  - Course overview
  - Succeeding in CSF

Administrative stuff

About the course

- Instructor
  - Brennon Brimhall, brimhall@cs.jhu.edu, 307 Malone Hall
- CAs
  - See course web page for details

Where to find stuff

- Course website: https://jhucsf.github.io/summer2025
  - Syllabus, schedule, lecture notes, assignments, etc.
  - All public course information will be here
- Slack https://jhucsf.slack.com/
  - Announcements
  - Discussion forum, Q/A: please post questions here!
- Canvas (accessible via MyJHU)
  - Not used

Syllabus highlights

- Please read the syllabus carefully:
  https://jhucsf.github.io/summer2025/syllabus/
- Highlights:
  - Grades: 55% homework, 39% exams, 6% participation
  - (Probably) 6 assignments, mostly programming based, expect them to be challenging!
  - Late policy: you have 60 late hours to use as needed
  - Three exams
    - Exams will be take home and available for 48 hours
    - You are expected to take them in one sitting
    - Will focus on recently-covered material

Pair assignments

- For most/all assignments, you may (optionally) work with one partner
- If you feel your partner isn't making an adequate contribution, you can finish the assignment on your own and turn it in individually
- You may not use your partner's lack of contribution as an excuse for not finishing the assignment

Participation

- What counts as participation?
  - What officially counts:
    - Participation in clicker quizzes in class
  - Also valuable, but won't officially count:
    - Activity on Slack (asking questions, answering questions)
    - Attending office hours
    - Reviewing lecture recordings
- I would like to see *reasonably consistent* participation
  - You can make up a missed day of class by attending a guest lecture

Academic integrity

- Please read the academic integrity policy in the syllabus carefully
- Had multiple issues with students last summer
- Highlights:
  - Follow the CS Academic Integrity Code: https://www.cs.jhu.edu/academic-integrity-code/
  - Generative AI is not permitted without prior written consent
    - If you want to use GenAI, please send me a note with a proposal and we can discuss
    - Note: a research emphasis for me is detecting and tracing generative AI output back to the individual user
  - Homework assignments
    - Individual: code sharing is not allowed
    - Pair: you can work with one partner
  - Exams are (obviously) individual effort
  - Violations of academic integrity will be reported to the Student

Academic integrity (continued)

To be considered for credit, your work and your partner's:

- Must consist of original code, written by you
- Must be a "clean room" implementation (i.e., you didn't use someone else's code as a reference)

We need to know that you can create a working implementation from a specification, on your own

Code generated by AI tools (including but not limited to ChatGPT, Github Copilot, etc.) will not be considered original work, and submitting such code is a violation of academic integrity

Public solutions

I am aware that there are solutions to CSF assignments posted publicly.

Please resist the temptation of copying code from these.

Because these projects are usually posted by students who have completed CSF, they are in my pool of previous submissions, meaning that they will come up in a code similarity comparison.

Class meetings

- Typical class meeting: lecture/discussion, peer instruction questions, occasional group activities, discussion of current assignment, time for free-form Q&A
- *Do the reading in advance!*
- Come prepared to actively engage with the material!
  - Learning is not passive
  - More productive class time $\rightarrow$ better outcomes
  - Ask questions!

Peer instruction

- How peer instruction works:
  - Slide with a multiple choice question
  - Answer individually on Slack, discuss with peers, then answer again
  - Shown to improve outcomes!
  - Questions may be challenging
  - Graded for participation only
- You may have done this in other courses

Peer instruction etiquette

- Be respectful:
  - Let everyone participate
  - Don't put down anyone else's ideas
- Work together and think carefully about the question!

First clicker quiz!

Clicker quiz omitted from public slides

Computing requirements

- All assignments will be done using x86-64 Linux
- Autograders will use Ubuntu 22.04
- **You will need an x86-64 Linux development environment!**
- Recommendations:
  - Ugrad machines (different version of Linux, but should work fine)
  - Run Linux on your laptop or PC
  - Run Ubuntu 22.04 using WSL2 under Windows (great option!)
- I'm not aware of any way to set up a usable development environment on an M1 Mac
  - VS Code and remote SSH to ugrad is a good option

Course overview

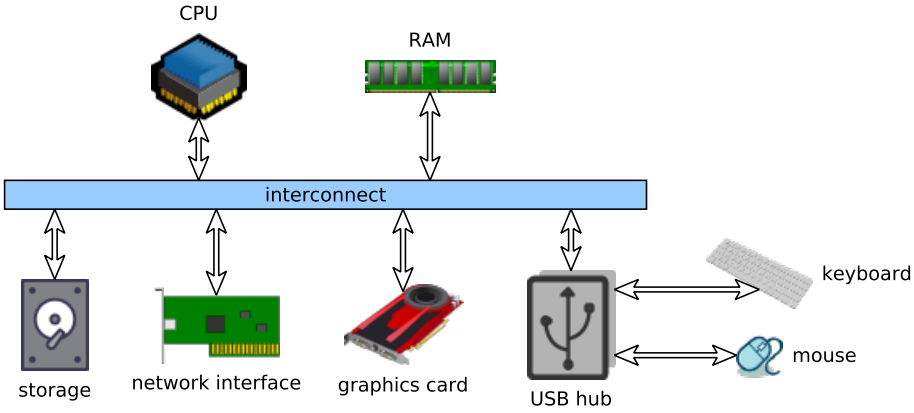
What the course is about

- Course is about *computer systems* from the *programmer's perspective*
- Computer system = hardware + software
  - Much of our concern is the interaction between hardware and software how they work together

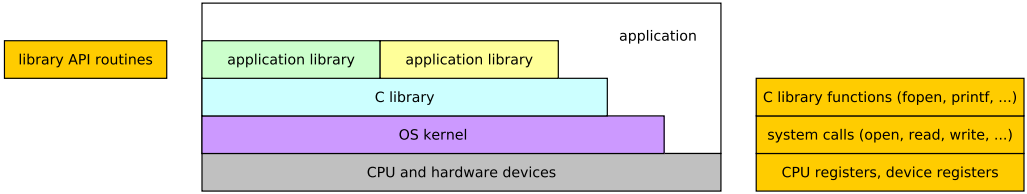
Goals of course

- "Deep" understanding of how computers work (down to hardware)
  - OS and runtime library interfaces
  - Machine-level ISA / assembly language
  - Processor features
  - Operating system features
- Apply this understanding to...
  - Optimize application performance
  - Avoid pitfalls such as security vulnerabilities
  - Take full advantage of the computer's and operating system's capabilities

A computer system (hardware)

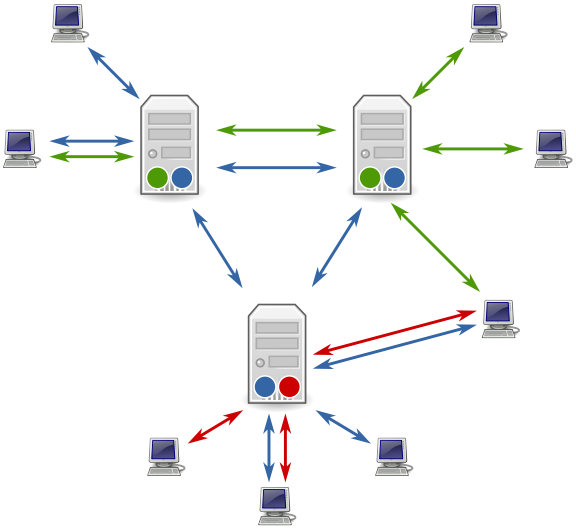


A computer system (software)



- Your application program is supported by lower layers of software and hardware
- Each layer provides an interface to the layer above

A computer network



Computer networks allow your program to communicate with peer systems.

Thanks to the global Internet, the peer systems could be anywhere on earth!

Succeeding in CSF

Succeeding in CSF

- We have a substantial amount of material to cover
- The work you do in CSF is (likely) different than what you've done in previous CS courses
  - Low-level data representation, assembly language, cache simulation, threads and concurrency, etc.
- You will be challenged
- You want to succeed, and I *want* you to succeed
- Let's discuss how to make this happen

Recommendation #1: start early, work steadily

How you manage your time is important in every course, but it's particularly important in CSF.

Scenario: you have a programming assignment to do. You have 2 weeks. We've indicated that it will be challenging.[1]

Time management option 1

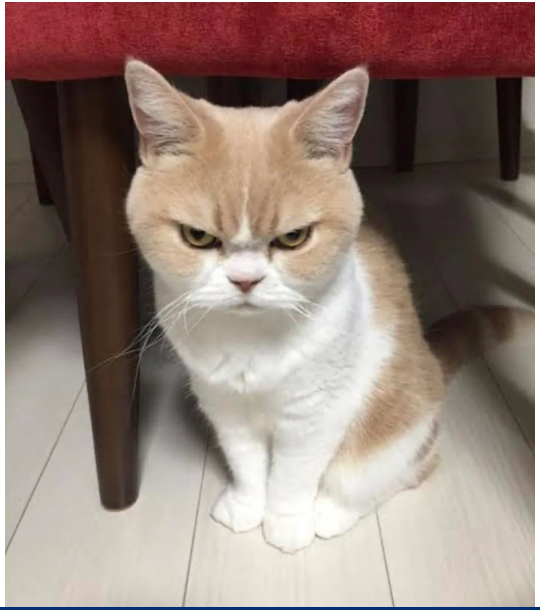
You wait until the day the assignment is due to start the assignment.

You get stuck and need help. Office hours are extremely busy.

You submit the assignment, but it's not fully functional.

You get a low score on the assignment.[a]

[a] https://www.reddit.com/r/aww/comments/gixv7e



Time management option 2

You start the assignment the day it is released.

You encounter an obstacle, but there's plenty of time to ask questions on Slack and get help in office hours.

You get all of the functionality working and have a week to spare.

Now you can chill.[a]

[a] https://unsplash.com/photos/orange-persian-cat-sleeping-9UUoGaaHtNE

The importance of starting early

It is completely expected that you will have questions about how to approach aspects of each assignment.

It is also expected that you will need time to think about how to best approach the assignment.

Starting early ensures that you are able to ask questions and have time to think and plan.

Time pressure can lead to poor decisions

In nearly every conversation I've had with students who violated academic integrity by plagiarizing someone else's code, they've mentioned being close to a deadline and panicking as the main factor.

Which leads us to...

Recommendation #2: communication

If you are concerned that for whatever reason you don't think you will complete an assignment, *let me know*.

I am generally willing to be flexible on a deadline if it means you will be able to succeed.

However: I want to see you actively making an effort to get things done. For example, I'm less excited about giving you an extension if it's apparent that you waited until the last minute before starting. If I've seen you actively asking questions, coming to office hours, asking good questions on Courselore, etc., I'm more willing to be flexible.

Flexibility

I try to provide substantial flexibility on deadlines. I know that everyone gets overloaded from time to time.

Late hours: everyone gets 60 late hours to use for assignments with very few restrictions. We do ask you to let us know (DM) if you plan to use more than 24 late hours on a submission.

But...flexibility is still possible even if you've used all of your late hours. Do **not** violate academic integrity because you are out of late hours and have a looming deadline. Let me know if you are in this situation.

Overall: as long as you are making a good faith effort to make progress in a timely manner, I will do my best to support you.

Recommendation #3: engagement

In my opinion, the whole point of in-person education is to be able to interact directly with other people.

Don't avoid this!

You should

- Come to class and participate!
- Attend my office hours
- Attend CA office hours
- Ask questions on Slack
- *Answer* questions on Slack

Next time...

Data representation (binary numbers, machine data types, addresses and memory)

Acknowledgements

Slides adapted from materials provided by David Hovemeyer.