

Today's plan

- Introduction
 - Course information
 - Syllabus
 - Logistics
 - Class format
- Class interactions
 - Your questions
 - Keys concepts/messages
 - Recap discussion
- Class exercises
 - Linux bootcamp

Course information

- Website: <https://jhu-ip.github.io/cs220-sp21>
 - Slides, videos, assignments, projects, office hours, and links to other resources
- Piazza: <https://piazza.com/jhu/spring2021/en601220>
 - Q & A platform, and announcement
- Blackboard: <https://blackboard.jhu.edu>
 - Links to the course website, Piazza page, and **Zoom sections**
- In other words, if you don't know where to find the course information, go to Blackboard.

Syllabus

- You should read it carefully
- A brief but important notes:
 - Coding (61%): assignments 31% + projects 30%
 - Code review (0%), but required
 - Absolute grading, not curved
 - total 4 late days for coding assignments
 - at most 2 late days can be used for one coding assignments
 - 30-minute grace period with 10% penalty
 - Start as early as possible, and reach out to us the sooner the better
- This class takes a flipped-classroom approach. You are expected to discuss your questions with us, demonstrate your implementation to us during class time.

Logistics

- Join Piazza, if you haven't joined (access via Blackboard)
- CS Account
 - If you have one, use it
 - If you don't, request a permanent one from CS IT if you are CS major/minor
 - Otherwise, send me a private post on Piazza with your Hopkins ID
- GitHub Account, follow instructions on this page:
<https://jhu-ip.github.io/cs220-sp21/docs/github/>
 - Between this Wed and Thur, you will receive an invitation to our GitHub organization.
 - Do remember to accept the invitation; otherwise, you will not be able to access the starter code and your personal repo
- Office hours survey - complete this survey (check the welcome email): <https://www.when2meet.com/?10804976-JeDMB>

Class format

- No “synchronous” lectures
- “Course Materials” page will be updated weekly
- Before coming to the class, you must
 - finish previous in-class exercise, watch videos, do recap questions, and prepare your own questions
- During class time,
 - I will invite volunteers (1-2) to demonstrate their implementation of the previous exercise
 - highly encourage to volunteer (it's a cheap failure)
 - the more feedback you get, the better and faster your learn
 - if no one volunteer,
 - I will be disappointedly showing my implementation, and
 - I will ask you to explain my coding logic
 - if it happens, I will randomly select 2-3 students (mostly likely those speak less) for this additional task

Class format

- During class time,
 - I will invite volunteers (1-2) to demonstrate their implementation of the previous exercise.
 - Next, I will **very briefly** recap the key concepts of the videos/slides you've gone through
 - Then, I will divide you into groups, each group will
 - discuss the questions that you've prepared, and
 - the recap questions in a breakout room
 - one of the group might discuss my coding logic in the main room (depending on if we have volunteers)
 - I will go into each room to discuss the questions you have prepared with you group
 - We will join the main room and summarize the answers together.
 - The time left is for you do the in-class exercise.

Class interactions

- Your questions
- Very brief recap
 - Course goal: write larger, complex, correct C and C++ programs by practicing, practicing, and practicing...
 - Short-term lazy vs long-term lazy: extra work up front that can reduce total time spent
 - Linux: a powerful system, the best for programmer
 - '~', '.', '..': home, current, parent directory
 - Remote access: use 'PuTTY' on windows or 'ssh' on Linux/Mac
 - I recommend to use WSL on Windows. Then, you essentially use Linux. <https://docs.microsoft.com/en-us/windows/wsl/install-win10>
 - 'pwd', 'ls', 'cd', 'mkdir', 'less', 'mv', 'cp', 'mv'
 - Use 'man' if you forget how to use a command
 - Use 'Tab' for auto completion

Recap questions

- ➊ What is the difference between short-term and long-term lazy?
- ➋ What is the 'ssh' command to connect to the ugrad machine?
- ➌ What are the commands to move, copy, and remove a file on a Linux machine?
- ➍ What should you do to learn C and C++ faster and better?
- ➎ What will we do during the class time?

Class exercises

Linux bootcamp: <https://cli-boot.camp/>