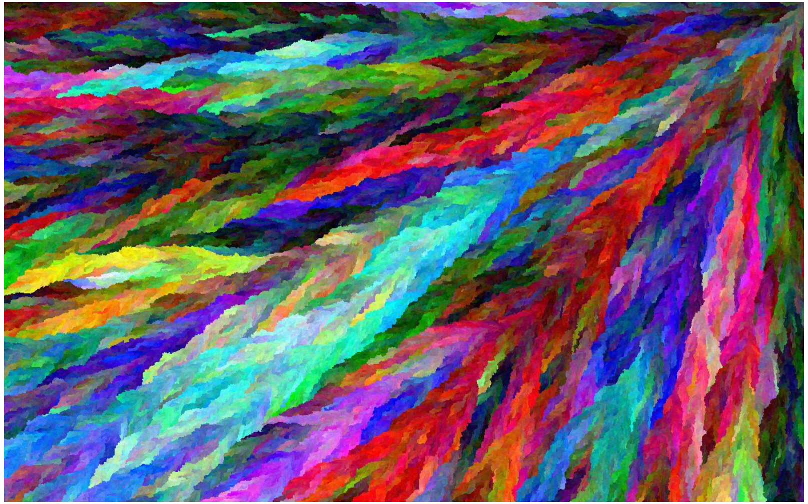
A Note for those Stumbling on these Slides (Hi!)

- These lecture slides are not intended as written reference materials.
  - Just reading them probably won't be very educational.

- Best used in combination with webcasts and source code references.
  - See (Video) and (Code) links under each lecture: http://datastructur.es



CS61B: 2020

Lecture 1:

- Introduction
- Course Logistics
- Hello World

61B Overview

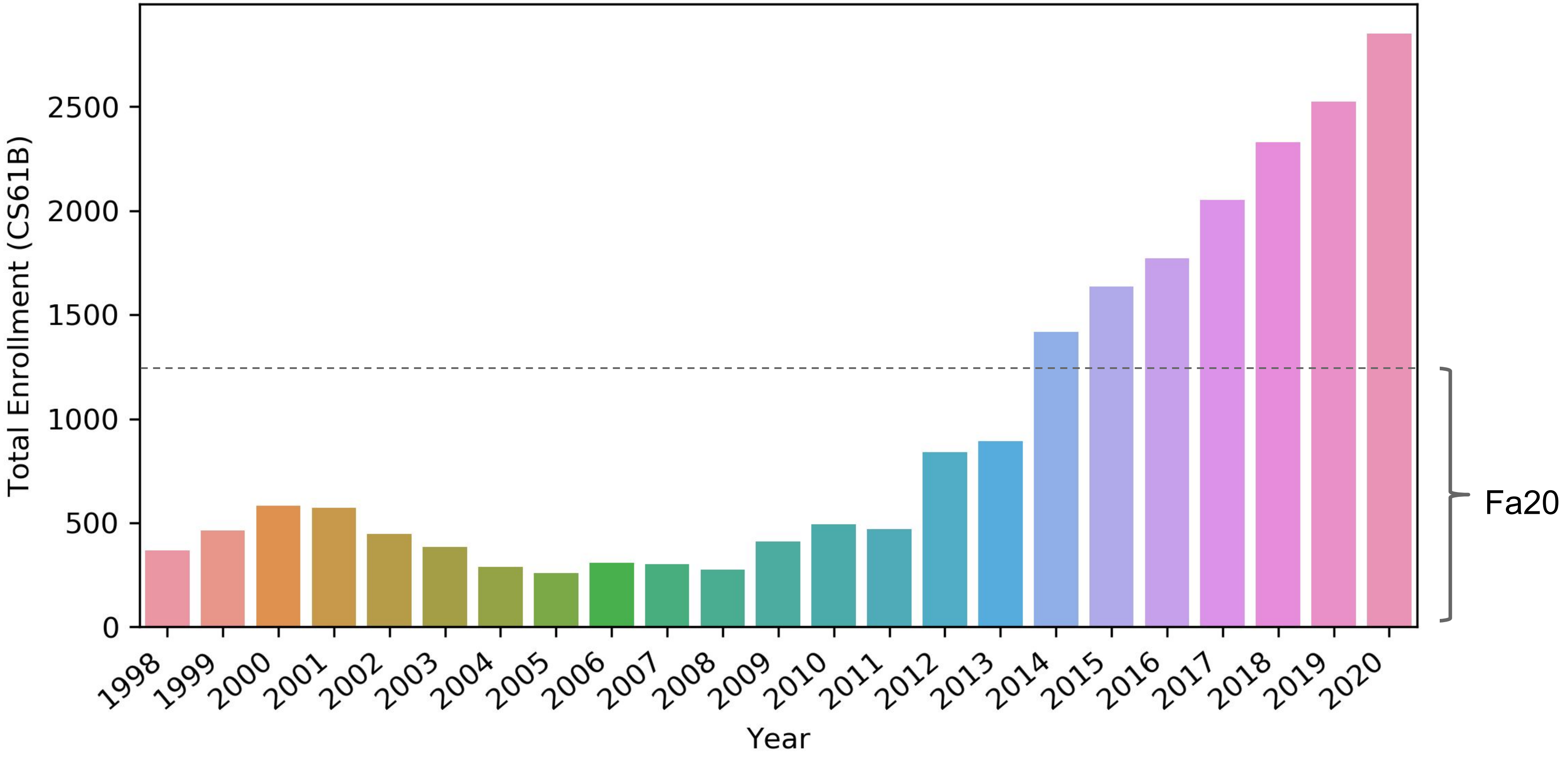
What is 61B about?

- Writing code that runs efficiently.
  - Good algorithms.
  - Good data structures.
- Writing code efficiently.
  - Designing, building, testing, and debugging large programs.
  - Use of programming tools.
    - git, IntelliJ, JUnit, and various command line tools.
  - Java (not the focus of the course!)

Assumes solid foundation in programming fundamentals, including:

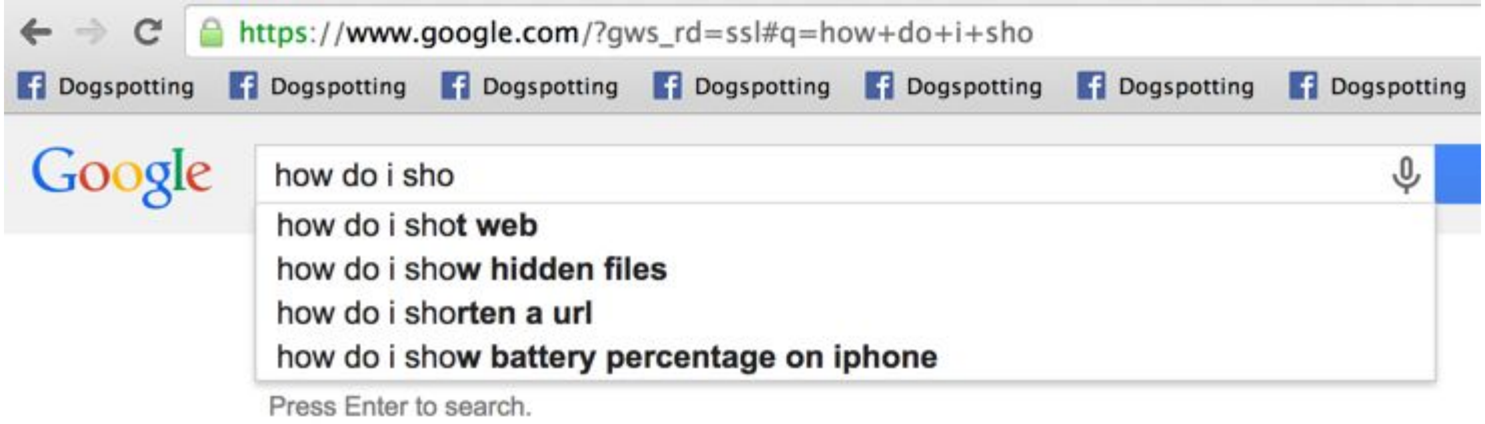
- Object oriented programming, recursion, lists, and trees.

Why Study Algorithms or Data Structures?



Why Study Algorithms or Data Structures?

Daily life is supported by them.



https://www.google.com/?gws_rd=ssl#q=how+do+i+sho

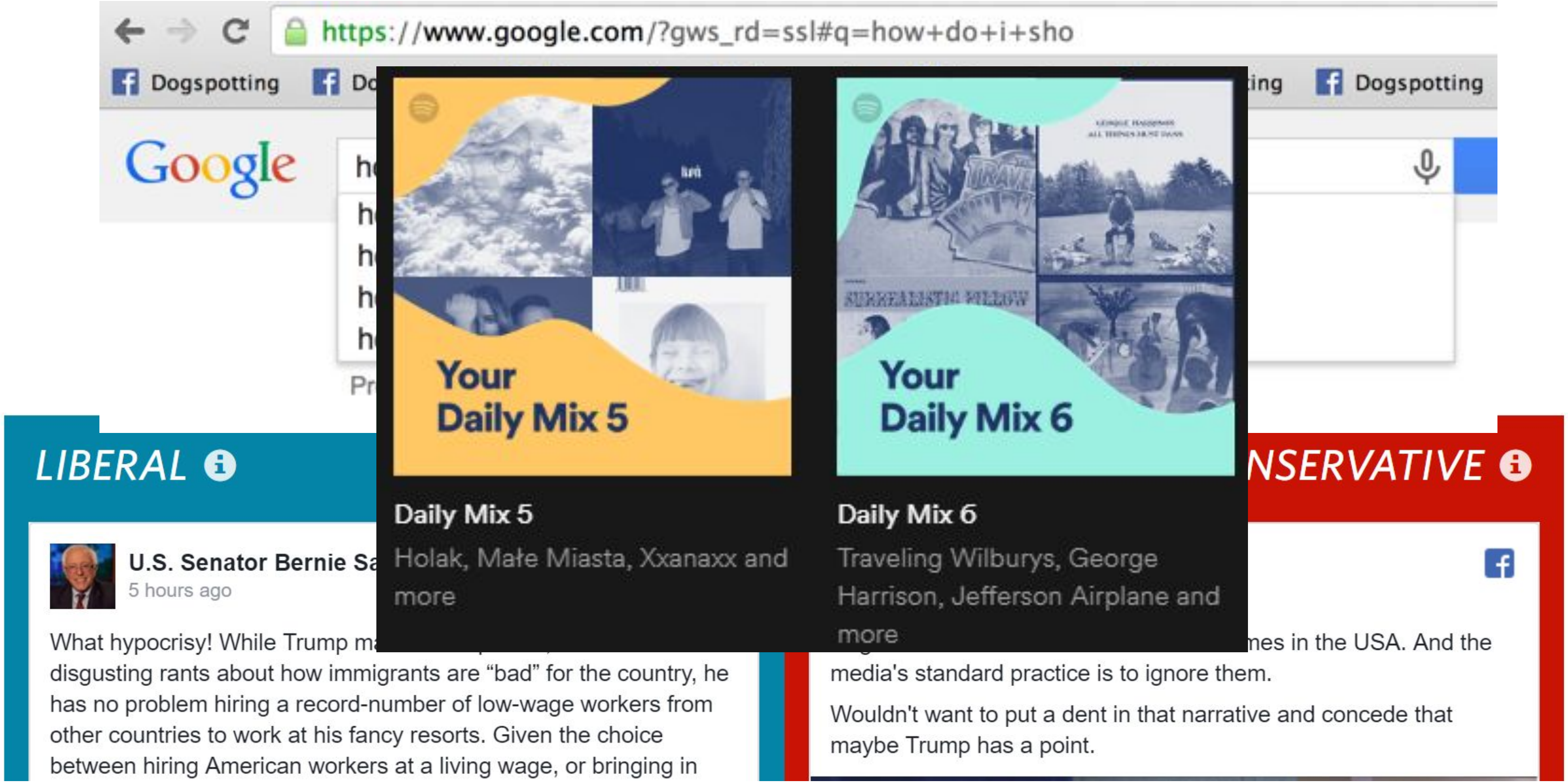
how do i sho

how do i shot **web**
how do i show **hidden files**
how do i sh**orten a url**
how do i show **battery percentage on iphone**

Press Enter to search.

Why Study Algorithms or Data Structures?

Daily life is supported by them.



Daily Mix 5

Holak, Małe Miasta, Xxanaxx and more

Daily Mix 6

Traveling Wilburys, George Harrison, Jefferson Airplane and more

U.S. Senator Bernie Sa

5 hours ago

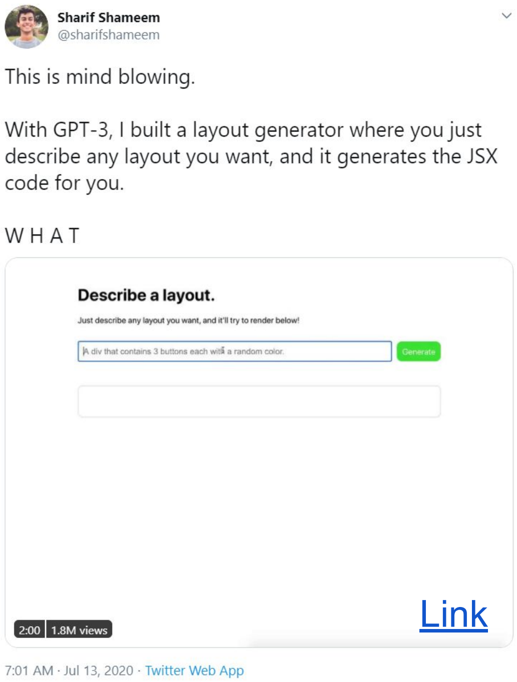
What hypocrisy! While Trump ma disgusting rants about how immigrants are “bad” for the country, he has no problem hiring a record-number of low-wage workers from other countries to work at his fancy resorts. Given the choice between hiring American workers at a living wage, or bringing in

mes in the USA. And the media's standard practice is to ignore them.

Wouldn't want to put a dent in that narrative and concede that maybe Trump has a point.

Why Study Algorithms or Data Structures?

Major driver of current progress (?) of our civilization.



Sharif Shameem
@sharifshameem

This is mind blowing.

With GPT-3, I built a layout generator where you just describe any layout you want, and it generates the JSX code for you.

W H A T

Describe a layout.

Just describe any layout you want, and it'll try to render below!

A div that contains 3 buttons each with a random color.

[Link](#)

7:01 AM · Jul 13, 2020 · Twitter Web App

Why Study Algorithms or Data Structures?

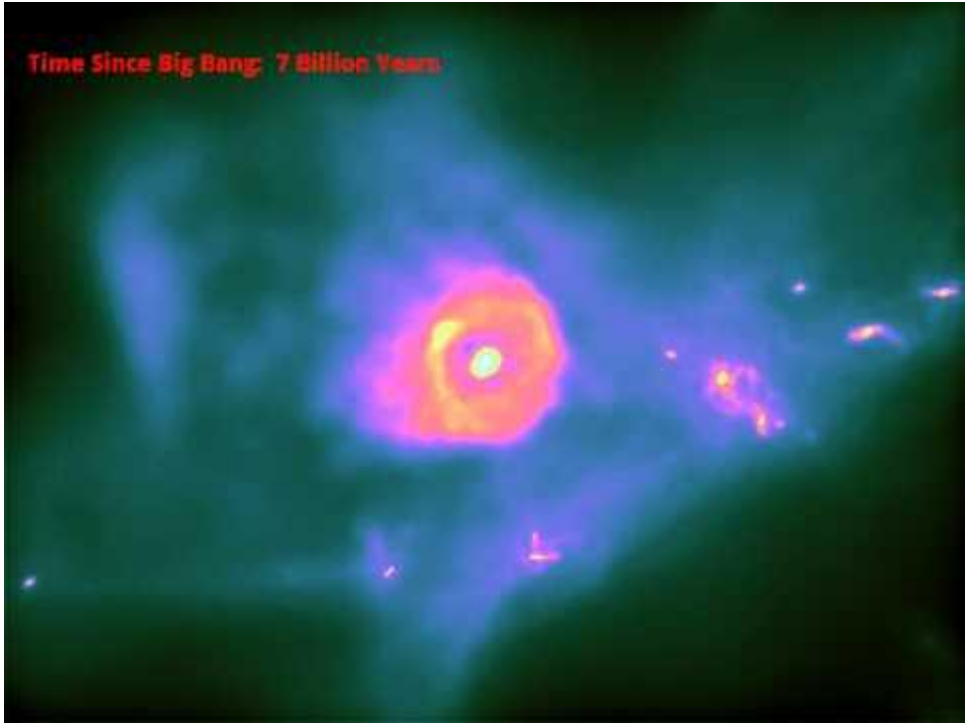
To become a better programmer.

> *“The difference between a bad programmer and a good one is whether [the programmer] considers code or data structures more important. Bad programmers worry about the code. Good programmers worry about data structures and their relationships.” - Linus Torvalds (Creator of Linux)*

Being an efficient programmer means using the right data structures and algorithms for the job.

Why Study Algorithms or Data Structures?

To understand the universe. Science is increasingly about simulation and complex data analysis rather than simple observations and clean equations:



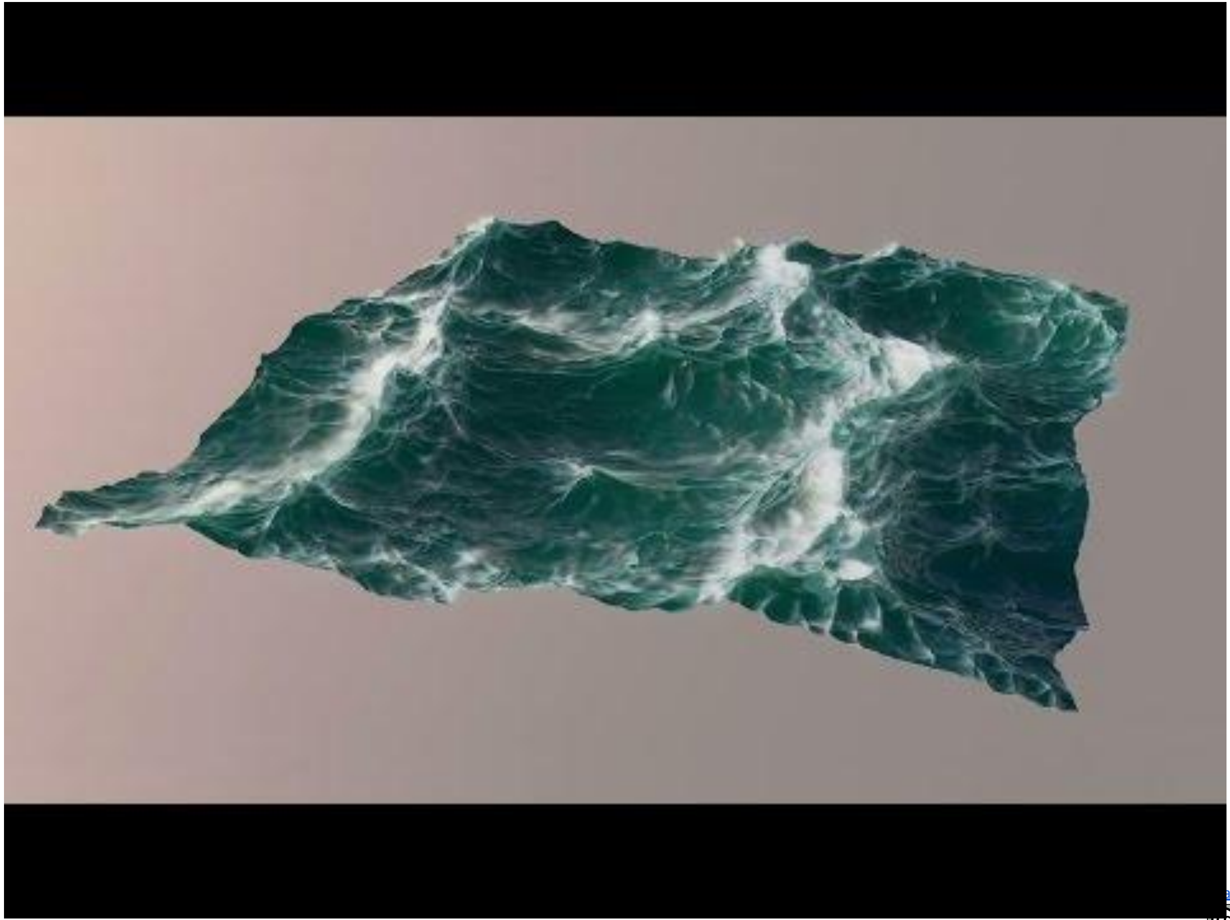
Why Study Algorithms or Data Structures?

to

create

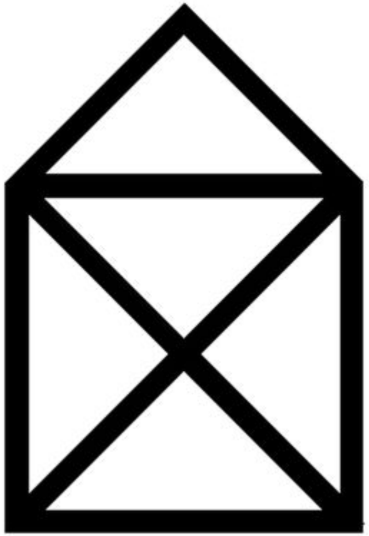
beautiful

things

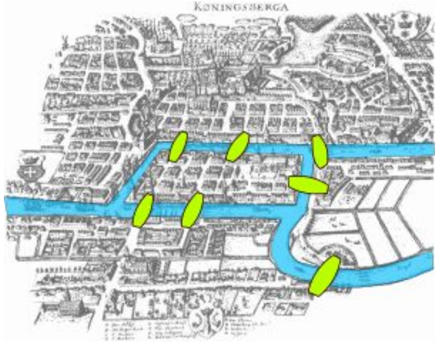


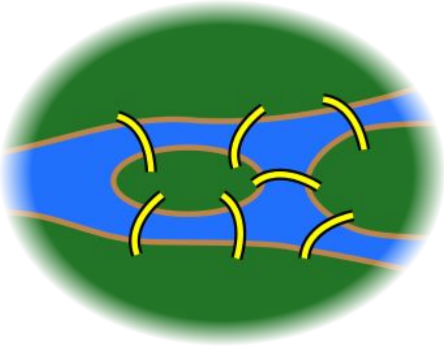
Why Study Algorithms or Data Structures?

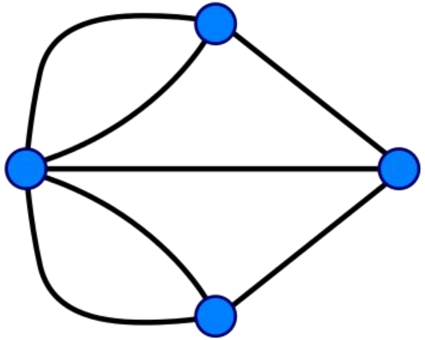
As an end unto itself.



Possible to draw without picking up pencil
or going back over any lines.







Impossible.

Question for You

What do you hope / expect to learn from this class? Why are you taking it?

- Better at technical interviews.
- Why useful for jobs? 61B covers topics like “hashing”, “search trees”, “binary search”, heaps”, “pqs” AND it also covers software engineering.
- Be better at programming.
    - Manage large projects (proj 2 and esp 3).
- **Break down big tasks into small tasks.**
- Learn about algorithms (last 5 weeks or so).
- Make programs more efficient.
- Need it to major in CS.
- Go surfing with Josh (me) someone said in chat. OK let’s do it.
    - I’ve always wanted to have a beach day for 61B. Probably not this semester, because of the deadly pandemic, but next time!
- It’s fine.

Who Are You?

Freshman: raise hand in zoom: 62

Sophomores ONLY: ~230

Juniors and Seniors ONLY: 180ish

Grad students ONLY: 80?

Who has taken a Java based class before? 220/739 (participants tab!)

- Demographic survey it was around 35% have taken some kind of Java.

Who Are We?

Instructor: Josh Hug (me) hug@cs.berkeley.edu 779 Soda (goodbye office)

GSIs:

- Tutors
  - Ayush Sehgal, David Shau, Jack Wang, Jacob Haddad, Shriya Nandwani
- Part time
  - Ada Hu, Adel Setoodehnia, Allyson Park, Boren Tsai, Chirasree Mandal, Claire Ko, Eric tang, Eric Zhu, Ethan Mehta, Fatema Yasini, George Zhou, Henry Kasa, Henry Maier, Isabelle Zhou, Linda Deng, Robin Qiu, Romain Priour, Ryan Tseng, Sara Reynolds, Sarah Liu
- Full time
  - Akshit Annadi, Alex Schedel, Anjali Kantharuban, Arjun Sahai, Connor Lafferty, Itai Smith, Omar Khan, Shivani Kishnani, Sohum Hulyalkar

Who Are We (continued)?

Academic interns:

Kyle Yu

Shilpi Shah

Justin Chen

Saikumar Gantla

Hannah Moore

Matthew Yu

Xinyu Fu

John Lee

Olivia Huang

Michael Wang

Yuwen Zhang

Tanya Sarkin Jain

Jeffrey Fan

Taewoo Han

Daniel He

Curie Park

Justin Lin

Emi Tran

Ethan Wu

Jeffrey Shen

Emily Su

Helen Hoang

Violex Ming

Smeet Patel

Anish Gollakota

Shreya Khandewale

Pranav Sukumar

Amritansh Saraf

Aekus Bhathal

Michael Paller

Tiffany Mao

Kyle Zhang

Sean Kim

Hana Lee

Angela Chen

Kishore Kumar

Noor Mahini

Ashley Auduong

Megan Yu

Rohit Mehta

Seraph Woodbury

Tejvir Jogani

Connor Lien

Michelle Chen

Rohan Khandelwal

Ramanan Abeyakaran

Richard Chang

Shefali Goel

Kenneth Ian Shyle

Nishant Patwardhan

Kuhu Sharma

Suraj Pakala

Marcus Hong

Ella Schwarz

Srinidhi Sankar

Brandon Marin-Mares

Kaaviya Sasikumar

Reena Yuan

Alexandra Lu

Ahmed Baqai

Surya Sunkavelli

Who Are We (continued (continued))?

Nandini Singh

Emma Toon

Farhad Alemi

Elise Ong

Brianna Fan

Kavya Marrapu

Anik Gupta

Anne Nguyen

Tarun Amarnath

Dennis Mach

Abhiraam Wadekar

Tiffany Kim

Aniruddh Chennapragada

Boxi Fu

Amrita Rajan

Jonah Noh

Aniruddha Vangala

Aryan Agrawal

Deepak Ragu

Eve Lin

Sakshi Satpathy

Spencer Jenkins

Ryan Huang

Yuanhan Li

Shreyas Kompalli

Derwin Wu

Avyakth Challa

Anton Zabreyko

Vincent Hwang

Arin Chang

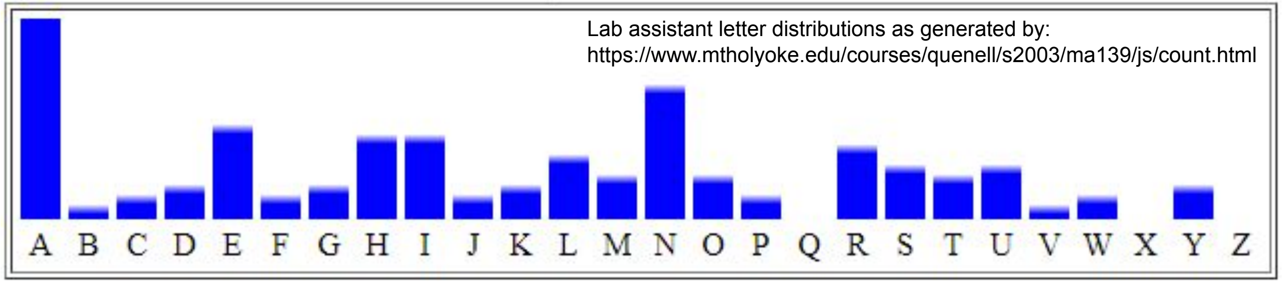
Joshua Blanchard

Ishaan Mauli Mishra

Tu (Alina) Trinh

Eugene Kim

Jonathan Yun



Lab assistant letter distributions as generated by:
https://www.mtholyoke.edu/courses/quenell/s2003/ma139/js/count.html

Learning Philosophy

The Manner in Which Learning Occurs (TMWLO)

TMWLO: A Small Minority

Lectures

- Introduction to new material.

Reading

- More thorough introduction.

TMWLO: The Vast Majority

Discussion Section and Study Guides:

- Practice with concepts. Often more theoretical.

Labs, Homework, and Your Own Experimentation

- Practice with tools, programming techniques, Java syntax, and algorithms and data structures.

Projects

- Similar to labs and HW, but larger and include a design component.

Course Logistics

Places to Get Information

Official Course Resources

- Course website: http://datastructur.es
- Lectures. Discussion. Lab. Office Hours.
- Ed Discussions: https://us.edstem.org/courses/979/discussion/
- Mini-Textbook: Obscurantism in Java https://joshhug.gitbooks.io/hug61b

Unofficial: Google, Stack Overflow, other programming courses on the web, various online documentation, etc.

Logistical Details

- For waitlisted folks: If you do project 0, I'll do what I can to get you in by week 4, but no guarantees.
  - Unlike past semesters, there is a chance we might not be able to accommodate everyone! Budget issues kept us from hiring as many TAs as usual.
- To sign up for a discussion section or lab, see https://us.edstem.org/courses/979/discussion/112067
- **Please post administrative issues to Ed or send an email to cs61b@berkeley.edu**
  - Please don't email me with such issues directly (sorry!).
  - 1200 students * 1 minute/student = 20 hours.

61B 3.1 - Course Structure

Phase 1: Programming Intensive Introduction to Java.

- Weeks 1-4.
- One browser-based programming HW (this HW0 is optional).
- Three labs to introduce you to various tools (starting this week).
- Two projects (proj0 and proj1).
- Midterm 9/23 at time TBD.

61B 3.1 - Course Structure

Phase 2: Data Structures.

- Weeks 5-10.
- Incredibly important and foundational material: Expect an CS job interview to lean heavily on this part of the course.
- Two programming HWs (HW1, HW2) and one exam-prep theory HW (HW3).
  - Applications and deeper insight into data structures.
- One large project (Proj2A/2B), due ~10/16 and ~10/23.
  - No autograder. You're responsible for your code's correctness.
- Standard labs implement data structures (e.g. hash table).
  - Challenge labs are more like hard puzzles using data structures.
- Midterm 10/28 at TBA.

61B 3.1 - Course Structure

Phase 3: Algorithms and Software Engineering

- Weeks 10-14
- Labs: Implement a basic algorithm (e.g. quicksort).
- Two projects:
  - Proj2C/2D: Implementation of an AI algorithm called A* and using your code from Proj 2A/2B/2C to implement browser-based Google Maps clone.
  - Proj 3: Build Your Own World: An open ended project where you build a 2D world with physics according to your own design.

See calendar at http://datastructur.es for more.

Labs and Discussion

Attendance for lab/discussion is not required, but can earn “pacing points” by attending discussion or lab.

- To earn pacing points, complete the lecture checkoff questions in time.
  - Will start week 2. We still don’t have our platform ready!
- See course website for details on pacing points.
  - tl;dr is that they can help you get up to a B-, but don’t help beyond that.

Labs:

- Lab always due by Friday at 11:59 PM. Full credit for ‘reasonable effort’.

HWs and Projects

3 homeworks.

- 2 are medium sized programming assignments. 1 is a theory homework.

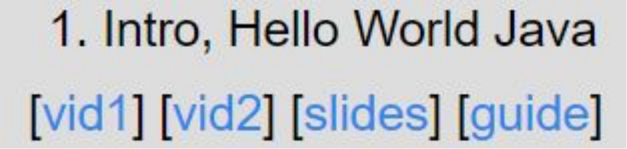
Projects:

- Projects 0, 1, and 2 are solo. Project 3 requires mandatory partner.
- Project 0 is light. Project 1 is significantly harder, but not huge.
- Projects 2 and 3 will be really time consuming and difficult.
- **All code on solo projects must be your own work.**
- Ok to discuss with others and help debug.
- Can earn gold points on projects for going above and beyond.
  - Gold points are special points whose value is worth more if your exam scores are lower. See course website for full details.

Weekly Surveys and Study Guides

Weekly survey due every Sunday.

- Check in on your progress and report attendance.
- Free points, but no late submissions allowed!
- Lowest four dropped.



1. Intro, Hello World Java

[vid1] [vid2] [slides] [guide]

Study guides for each lecture.

- HWs will not prepare you directly for many types of exam problems.
  - Use these **study guides** instead!
- Provides a brief summary of the lecture.
- Provides (usually) C level, B level, and A level problems for exam studying.
  - A level problems are usually hard enough that I anticipate TAs will have a hard time with them, so be nice!

Exams

- Closed note except you can bring cheat sheets.
- Will be pretty hard (60% medians).
- Showing improvement on final can boost overall exam score.
    - If your final is statistically better, it can replace one of your two midterms. See “supersession” under course website for details.

Exam dates:

- Midterm 1: **September 23rd**, time TBD. Will accommodate timezones.
- Midterm 2: **October 28th**, time TBD. Will accommodate timezones.
- Final Exam Times: **Dec. 16th**, 7:00 - 10:00 PM, **Dec. 17th**, 8:00 - 11:00 AM
- There will be **no alternate exams** (see exam replacement policy).
    - One exception: Direct conflict with both finals with another class.

Course Grade

Breakdown: 6,344 points total. Letter grade will be determined by your total.

- Midterms: 1600 points total.
- Final: 1600 points.
- Projects: 2120 regular points.
- HW: 480 points (160 points each)
- Lab: 384 points (32 points each)
- Weekly Surveys: 160 points (16 points each)

Plus extra credit for filling out pre-, mid-, and post- semester course surveys.

Plus gold points for going above and beyond on projects.

Plus pacing points for keeping up with the course.

Grades are not curved, i.e. they are not based on your relative performance. In past semesters, my grade bin cutoffs have not moved much at all.

- See http://fa20.datastructur.es/about.html for full details including grading bin cutoffs.

Lateness and Mentor GSIs

No late work will be accepted in the course. No hws/labs/projects are dropped.

- You should treat the deadlines seriously!
- If needed, use automated extension system to request extensions.
  - Up to 15 total slip days.
  - Maximum of 3 per assignment.
  - Don't expect similar extensions in other classes. 61B is not as strongly cumulative as classes like EE16A or CS70.

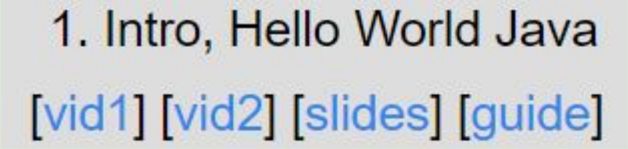
During week 2, you will pick a **mentor GSI**.

- If you do not pick one, one will be assigned for you.
- Mentor GSI will keep track of your progress and reach out (or have someone else reach out) if they feel like there are potential issues.

Course Pacing

We will start off very fast.

- Optional HW0 is out.
  - Intro to Java syntax.
  - Will take 1-4 hours.
  - Work with friends!
  - Recommended that you complete before your lab.
  - Strongly recommended that you complete by lecture Friday.
- Lab1 and Lab1 Setup are both available.
  - Lab1: How to use various tools.
  - Lab1 Setup: How to set up your home computer (maybe do before lab1).
- Project 0 released today. Due Friday Sep 4th (9 days from start of semester).
  - Start by Saturday if possible, especially if you're new to Java.
  - Exercises all the basic Java features.



1. Intro, Hello World Java

[vid1] [vid2] [slides] [guide]

Hello World

(See guide for link to the code I write today)
(Might be a little boring if you know Java already)

Java and Object Orientation

Java is an object oriented language with strict requirements:

- Every Java file must contain a class declaration*.
- **All code** lives inside a class*, even helper functions, global constants, etc.
- To run a Java program, you typically define a main method using
  `public static void main(String[] args)`

*: This is not completely true, e.g. we can also declare “interfaces” in .java files that may contain code. We’ll cover these later.

Java and Static Typing

Java is statically typed!

- All variables, parameters, and methods must have a declared type.
- That type can never change.
- Expressions also have a type, e.g. "`larger(5, 10) + 3`" has type int.
- The compiler checks that all the types in your program are compatible **before the program ever runs**!
    - e.g. `String x = larger(5, 10) + 3` will fail to compile.
    - This is unlike a language like Python, where type checks are performed DURING execution.

Your Reflections on Static Typing

The Good:

-

The Bad:

-

What’s Next

This week:

- HW0: Out now. Will give you a chance to explore Java basics on your own.
- Lab1: How to compile and run code. How to check out homework starter files and submit them. **If possible, do HW0 before lab!**
- Lab1 Setup (optional): How to compile and run code on your own machine.
- Project 0 (released today). Must be done solo.
- Next lecture: What all that public static blah blah stuff actually means.