Introduction

Introductions: Harry

Welcome! I'm Harry Smith (he/him/his)

- As a lecturer, my job is **teaching**—I'm here for you!
- Find me at sharry@seas.upenn.edu or in Levine 260
- Office Hours: Mon 10-11:15am & Tue 3:45-5pm
- Advising Hours: Tue 10:15-11:30am

About Harry

Undergrad	Here!
Grad School	Columbia
Academic Interests	Data Viz, CS Education
Height	6'1
Favorite Food Near Campus	Han Dynasty
Favorite Album of 2024	*Only God Was Above Us* by Vampire Weekend
Favorite Authors	Rachel Cusk, Elena Ferrante, Natalia Ginzburg
Favorite Video Game	*Disco Elysium*

10 Years Ago This Week

CIS 110 Permit—Harry Smith

Harry Smith <sharry@sas.upenn.edu>
to bjbrown

Fri, Aug 22, 2014, 6:58 PM

Hello Professor Brown,

My name is Harry Smith and I am a freshman this Fall. I am interested in taking your CIS 110 course in the 002 section for the Fall 2014 Semester. In order to do so, I need to obtain a permit from you. If you could provide me with information about how to obtain the permit or if prerequisite courses are necessary, I would be appreciative.

If some background information about myself and my academic past is necessary, then here it is: I am a student interested in majoring in Chemistry. I very much enjoy mathematics and I have completed high school mathematics through the IB HL Math level (which is approximately commensurate with BC Calculus) and received a 5 out of 7 in the course.

Thank you for the help,
Harry Smith

Introductions: Jessica

Hi! I'm Jessica Shi (she/her/hers).

- My email is jwshi@seas.upenn.edu.
- My office hours are Friday 1:30–2:30 p.m. in Levine 260.

About Jessica

Undergrad	MIT
Grad School	Here! (And Now!)
Academic Interests	Programming Languages & Human-Computer Interaction
Height	Not 6'1
Favorite Food Near-ish Campus	Tampopo
Favorite Cozy Mystery	*The Thursday Murder Club*
Favorite Jigsaw Puzzle	*UFO Victorian Ladies*

Introductions

We have so many excellent TAs this semester here to help you learn.

CIS 1100 001 vs. 002

This section (001) of the course is taught in **Java,** and the later section (002) is taught in **Python**

- In common:
  - No prior experience required!
  - All requirements satisfied!
  - Good Vibes
  - Harry

CIS 1100 001 vs. 002

This section (001) of the course is taught in **Java,** and the later section (002) is taught in **Python**

- Unique to this section:
  - Java
  - Jessica
  - A tried-and-true, well-worn curriculum

Who is CIS 1100 For?

🎉 🎉 🎉 *Everyone!* 🎉 🎉 🎉

Who is CIS 1100 001 For?

🎉 🎉 🎉 Everyone! 🎉 🎉 🎉

But especially for those who:

- already plan to take lots more CS at Penn
- like a more traditional lecture experience
- are CIS or Computer Engineering students

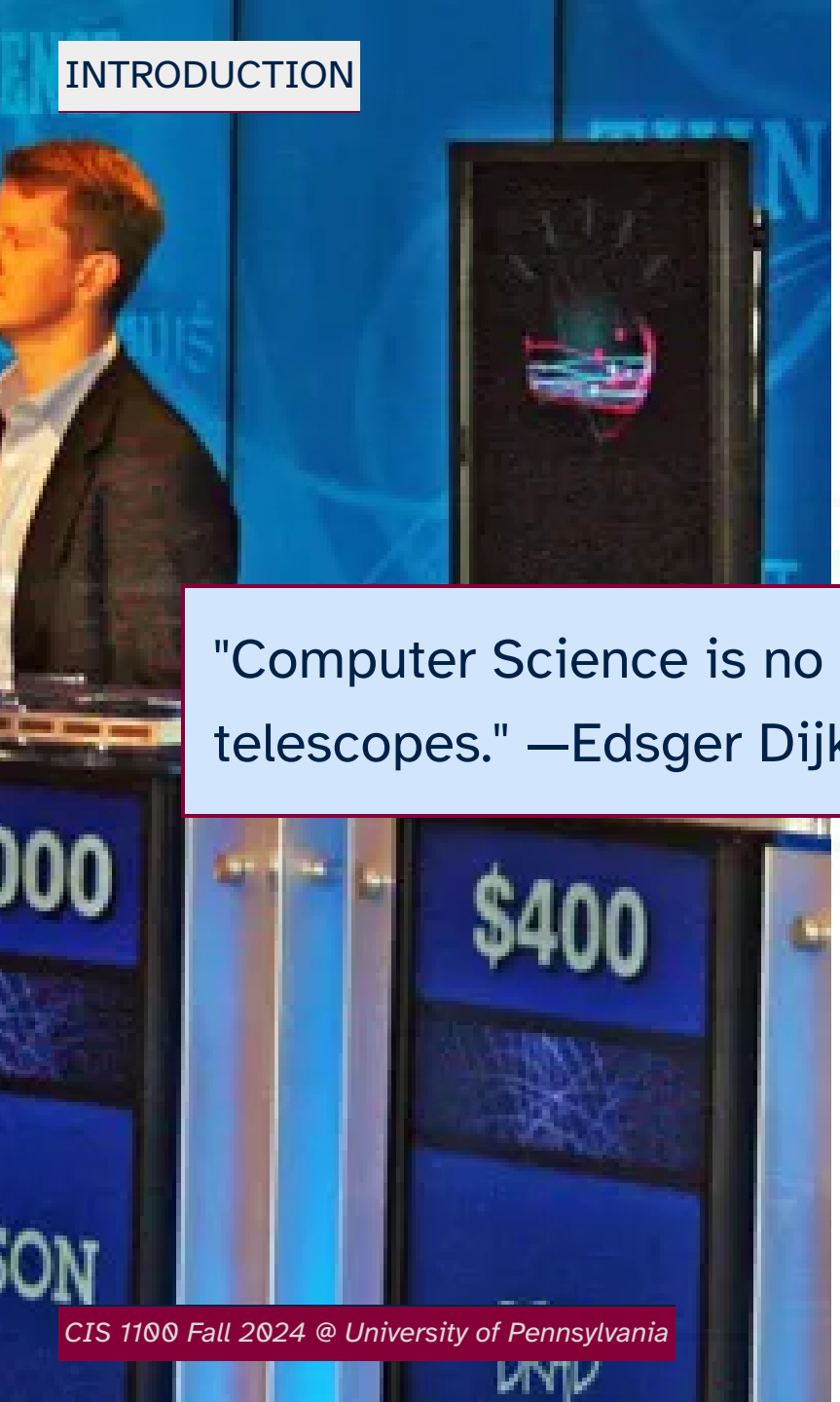
WHAT SHOULD I DO?

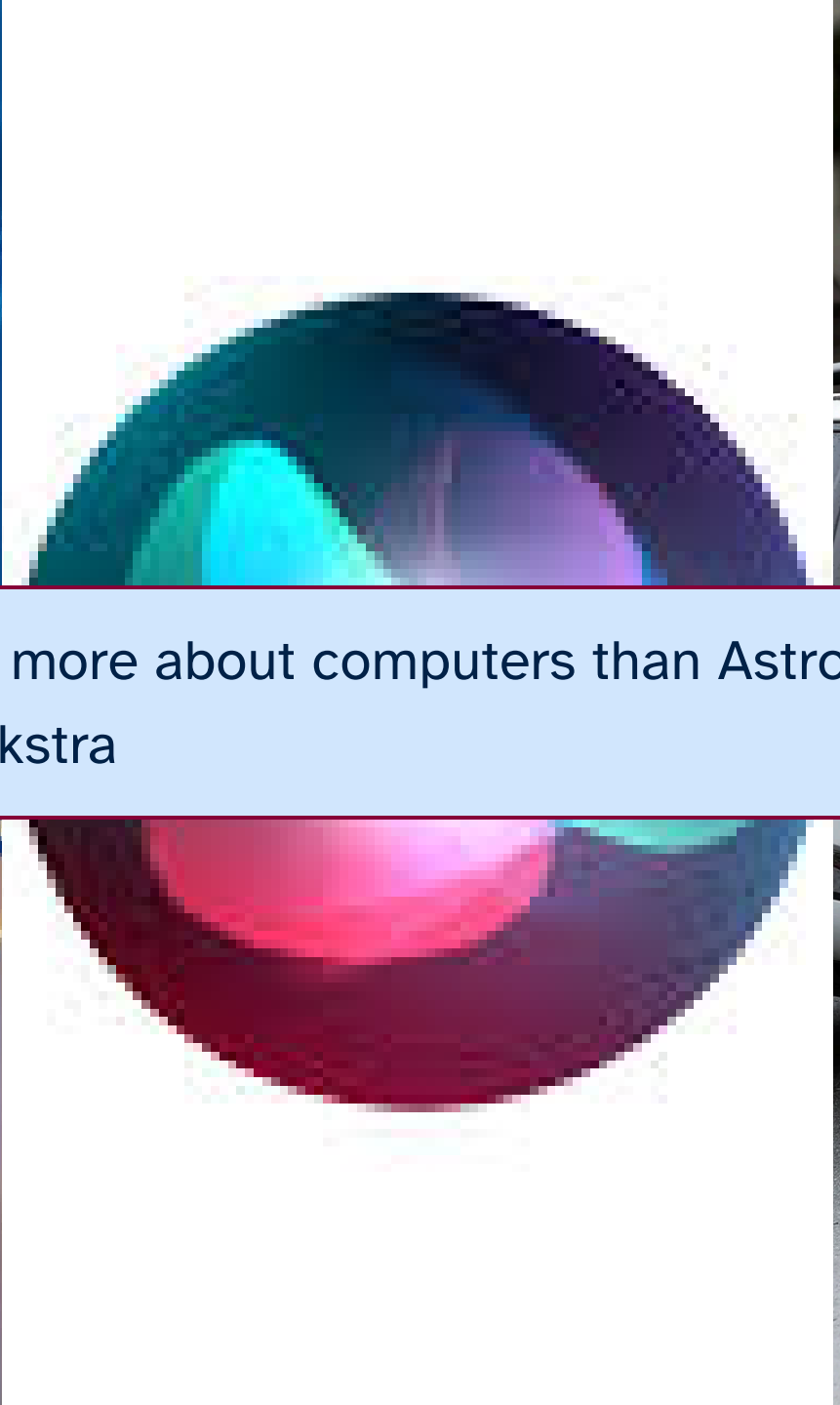
Take whichever section seems more interesting to you!

If you're indecisive:

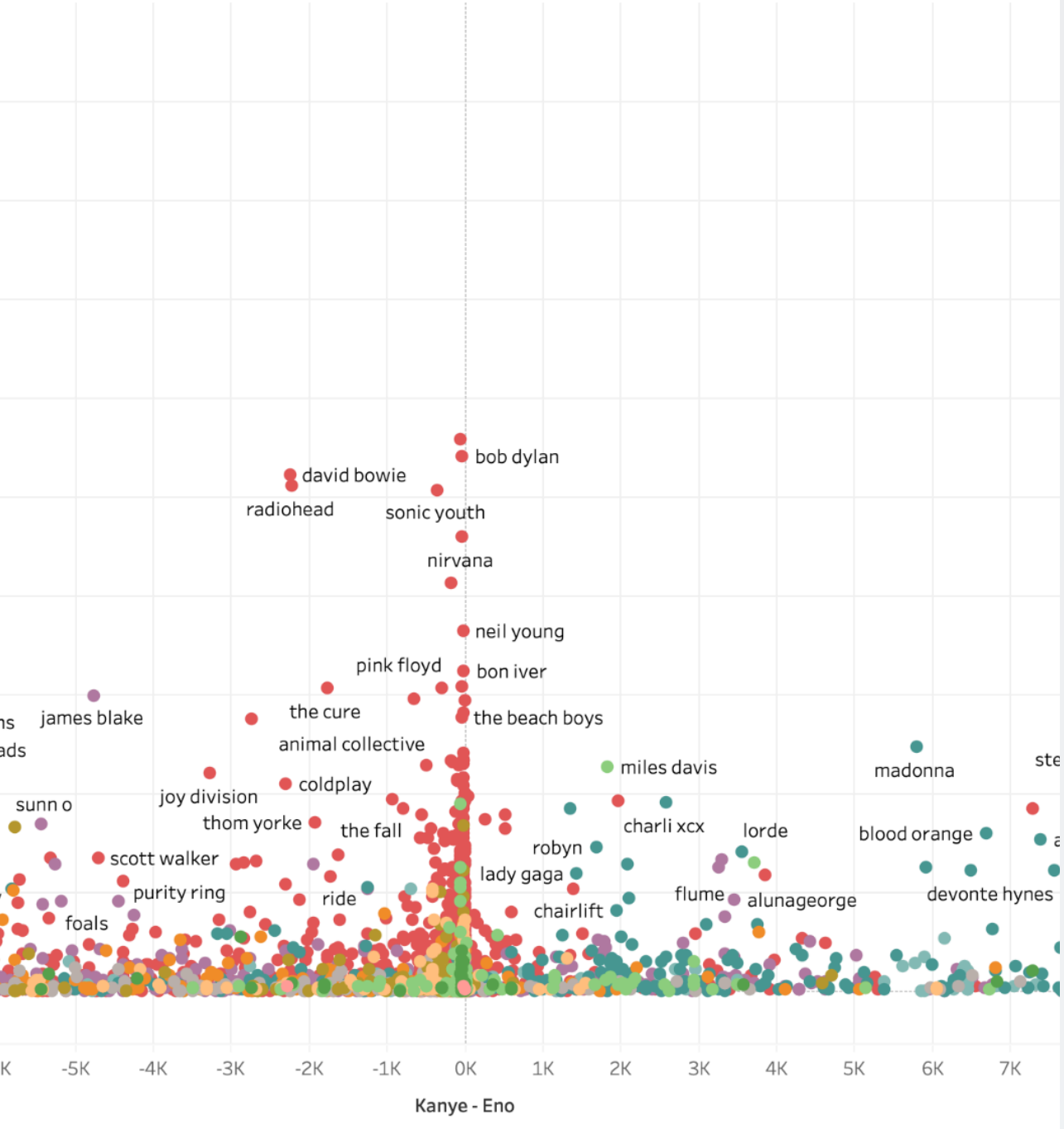
- attend both lecture sections. You have until Sep 11 until your first homework assignment is due.
- Talk to the TAs & instructors

What is CIS 1100?

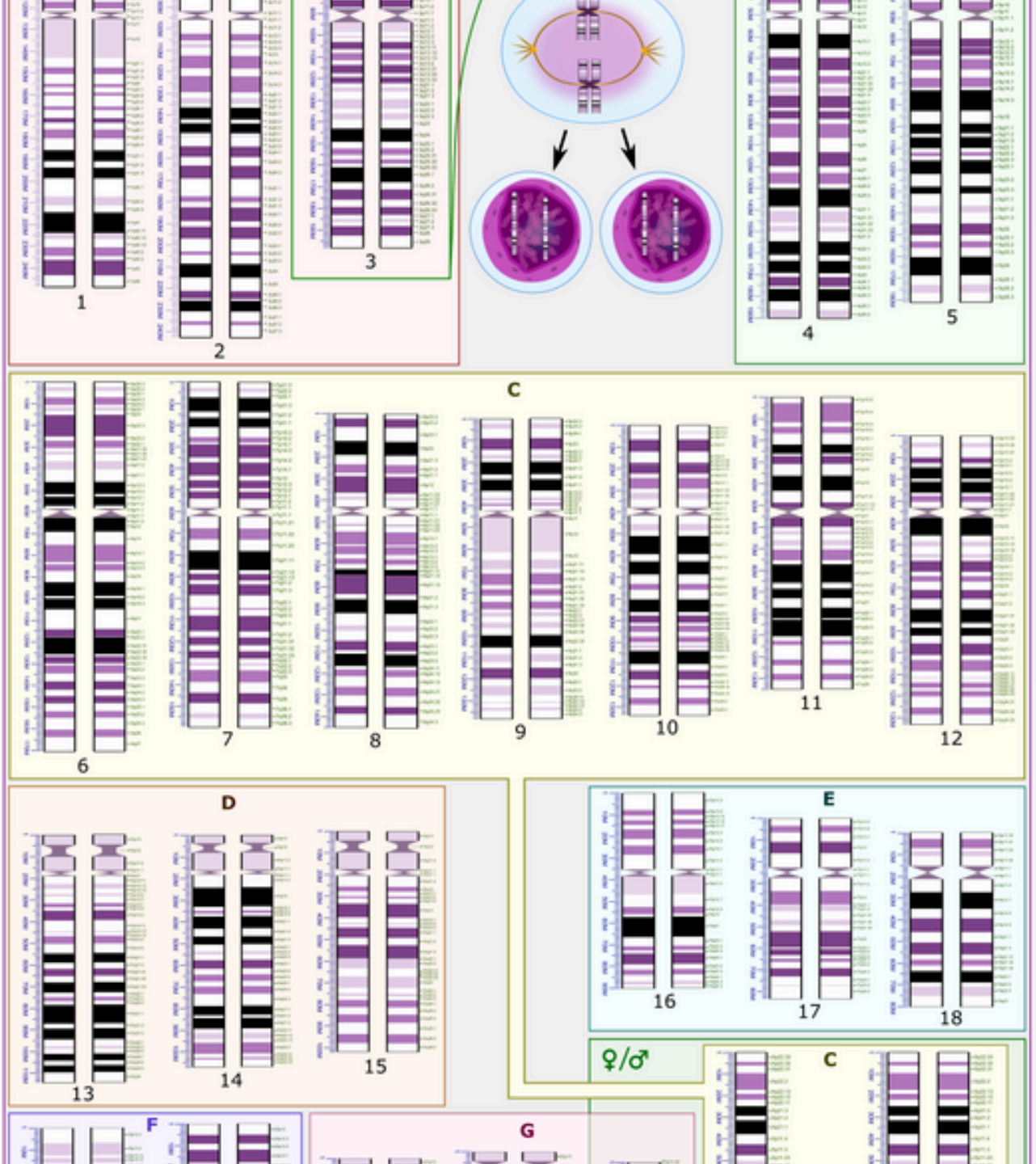




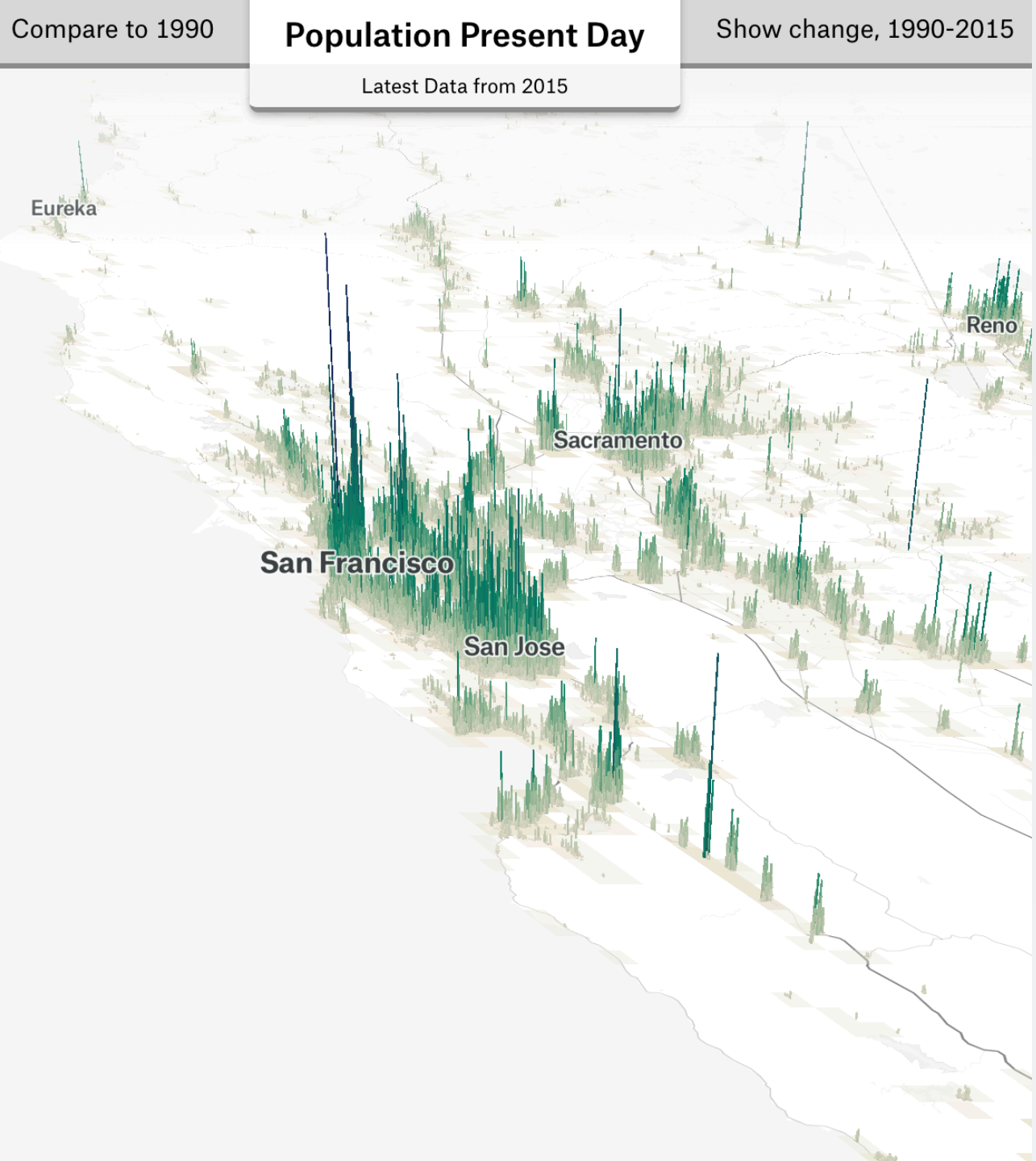
"Computer Science is no more about computers than Astronomy is about telescopes." —Edsger Dijkstra



Digital Humanities



Biology



Data Journalism

Guiding Principles

- Getting better at stuff requires PRACTICE.
  - To make sure that you are building your programming skills, we are trying to give you a lot of practice to make you better at programming.
  - This practice comes in several forms: check-ins, lecture activities, recitation, homework assignments
- In the next slides:
  - details about what we'll be asking you to do
  - rationale for why we think it's important

Lecture

- Three times a week.
- Lecturing, with additional live demos, Q&A, and polls.
- **Recordings:** Available on Canvas.
- You are expected to engage, ask questions, practice *metacognition*.

Recitation

- These are **mandatory** weekly sessions
- **Format:** Review, group work, TA interaction.
- You must be registered for **one** section
- Attendance & participation included in your final grade!

Assignment Types

- You will be assessed in various ways outside of lecture and recitation.
- There are four main types of assignments: 📝

Weekly Check-In Activities

- Short worksheets or small programming tasks.
- Complete them individually or in pairs.
- Submit and repeat as needed.
- Get guidance from TAs.

Programming Projects (Homeworks)

- One- or two-week assignments.
- Apply course material to real programs.
- Must be done individually.
- Seek TA assistance for understanding and debugging.
- Challenging, may take 10-15 hours.

Homework Policies: Late Submissions

- You have four "late days" for Homeworks.
- Each "late day" allows up to 24 hours delay.
- At most two late days per assignment.
- No submissions after the deadline without a late day. ❌

Homework Policies: Collaboration

- Homeworks must be completed individually.
  - We check this. There are consequences. Not fun.
  - (exception is for portions of HW assignments that we begin working on during lecture.)
- Internet (Google, StackOverflow, ChatGPT & LLMs) not acceptable.
  - reasonable to want some help, but these sources too easily give it all away
  - ask on Ed, during lecture, or in Office Hours if you need help

Why No AI??

We are skeptical of its usefulness in actual learning

- https://www.aisnakeoil.com/p/chatgpt-is-a-bullshit-generator-but
- https://www.aisnakeoil.com/p/gpt-4-and-professional-benchmarks

Will not help your overall grade and happiness:

- If you can’t explain your code in OH, we can't help you
  - This is different than being confused on a bug or with Java, this is ok
- Everything we do here is foundational for ALL programmers. Why spend 🕐 and 💵 on this course if you don't want to learn how to do the stuff in it?
  We give a lot of help in the class, come get help if you need it.

Midterm Exams

- Two exams during the semester.
- Held in class:
  - October 9 and November 20**
  - Wednesdays!
- Closed-book exams. 🚫📚
- Together, they account for 12% of your final grade.
- Test your Java fluency under time constraints.

Final Exam

- Scheduled during the official finals period. 🗓️
- Worth 12% of your final grade.
- Similar to midterms, assesses Java fluency.
- Date and time will be announced soon.

Tips for Success

Other quick tips for success:

1. Start your assignments as soon as they're released to you.
2. Come to Office Hours all the time, even if you don't have a question.
3. If you ever need support, either academic or otherwise, don't hesitate to reach out.

TA Resources

Sunday Review Sessions

- Weekly on Sundays! Time TBD
- Useful for exam review, interactive small group work

Code Reviews

- One of these is mandatory in the first month of the course
- After that, you can sign up for them as needed!
- Great for 1:1 time with a TA

Online Resources

- Course Website: cis1100.com
  - hw assignments, schedule, recitation info, course policies
- Canvas
  - class recordings
- Gradescope (coming soon)
  - assignment submission & grading
- Codio (use course token `north-alias`)
  - writing code

First Program: `HelloWorld.java`

We'll write a Java program that prints out the message `"Hello, World!"` when compiled and run.

Definitions:

- Java program
  - Code, written in the Java language, that specifies a series of instructions for a computer to execute
- Printing 🖨️
  - Displaying text to the screen
  - (not actually putting ink on a piece of paper)
- Compiling
  - A process that turns Java code (stuff that we write) into specialized instructions that a computer knows how to execute
- Running 🏃‍♀️
  - Executing the compiled instructions!

File Naming Conventions

When you create a new file in Codio, you'll have to name it in a special way to make sure it's run and recognized as a Java program.

- For our first program, we'll call it `HelloWorld.java`.
  - 🚨Make sure to use this name exactly!🚨
- Since the filename of our program is `HelloWorld.java`, our program should start with `public class HelloWorld`

```
public class HelloWorld {

}
```

The "Class"

- Further study required to really understand this term 😕
  - This is one of the (myriad) miseries of Java!
- For now, we'll say that the concept of a *class* is synonymous with that of a *program*.
  - A bunch of code that lives in a file that is identified with a particular name
  - In our first case, `HelloWorld`

The `main` method

Next up, we'll add some more jargon to our program.

```
public class HelloWorld {
  public static void main(String[] args) {

  }
}
```

This new stuff in the middle of the braces we wrote after the class declaration is called the "`main` method". It has its own curly braces which will contain all of the statements that actually give our program interesting behavior.

Finishing up the Program

```
public class HelloWorld {
  public static void main(String[] args) {
    System.out.println("Hello, world!");
  }
}
```

Here, we've added a *print statement* to our `main` method that tells the program to print out the message `"Hello, world!"`.

Print Statements

```
System.out.println("Your message goes here! Don't forget the quotes.");
```

- Writing `System.out.println` followed by a pair of parentheses allows you put a message in those parentheses to be printed out.
    - Make sure to put the text of the message between a pair of double quotes (these `"`).
- Remember to put a semicolon (`;`) at the end of the line, too!

Comments

These are used as notes that describe your program, contained within the program.

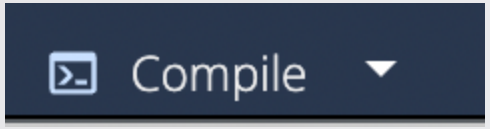
They are ignored by the computer and are meant only for other humans (you, your teammates) to read.

```
// this is a comment on one line, denoted by the double slashes

/* You can also write multi-line comments this way,
 * starting them with "slash-star" and ending with
 * "star-slash".
 */
```

Compiling and Running

To see what the program does, you have to *run* it! To run it, you must first *compile* it.



(click this button here!)

Compiling

README.md | .codio | HelloWorld.java | Compile

```
Welcome to Ubuntu 14.04.5 LTS (GNU/Linux 5.19.0-1028-aws x86_64)

 * Documentation:  https://help.ubuntu.com/
 *
 * Welcome to the Codio Terminal!
 *
 * https://docs.codio.com/develop/develop/ide/boxes/overview
 *
 * Your Codio Box domain is: rentminus-lasermayday.codio.io
 *
Last login: Tue Jul 25 01:23:37 2023 from 192.168.10.93
codio@rentminus-lasermayday:~/workspace$ javac -cp cis110.jar:junit-platform-console-st
andalone-1.3.2.jar:. *.java
codio@rentminus-lasermayday:~/workspace$
```

Successful!

Compiling

README.md | .codio | HelloWorld.java | Compile

```
codio@rentminus-lasermayday:~/workspace$ javac -
andalone-1.3.2.jar:. *.java
HelloWorld.java:3: error: cannot find symbol
    System.out.println("Hello, world!");
               ^
  symbol:   method println(String)
  location: variable out of type PrintStream
1 error
codio@rentminus-lasermayday:~/workspace$
```

Failure! The message reports `"1 error"`.

Compiling

```
codio@rentminus-lasermayday:~/workspace$ jav
andalone-1.3.2.jar:. *.java
HelloWorld.java:1: error: '{' expected
public class Hello World {
                      ^
1 error
codio@rentminus-lasermayday:~/workspace$
```

A Different Failure! (again, `"1 error"`).

Running

Once you have a successful compilation (no errors reported), you can run the program by typing **into the terminal window** `java HelloWorld`.

```
codio@rentminus-lasermayday:~/workspace$ java HelloWorld
Hello, world!
codio@rentminus-lasermayday:~/workspace$
```

See the message that got printed?

For You to Try

1. Try changing `"Hello, World!"` to something else—but remember your quotation marks!
2. Try adding another line of `System.out.println();` with a different message.
3. Replace the first `System.out.println()` with `System.out.print()`—what's the difference?