

Before we get started...

In class and in lab, I expect you to contribute to an environment that supports everyone's learning.

- 1 Please raise your hands to ask or answer questions
- 2 No phones, no headphones; use computers but only for classwork.
- 3 No food in class or lab!

Hello, computer



We use computers every day as electronic *black boxes* that do amazing things by

collecting,

storing,

retrieving, and

transforming *data*.

“Many people think of data as numbers alone, but data can also consist of words or stories, colors or sounds, or any type of information that is systematically collected, organized, and analyzed...”

D'Ignazio & Klein, *Data Feminism*, 2020



James Murray compiling
the *Oxford English
Dictionary*, c. 1928.

Computers only do very basic things.

Numerical calculations:

Add

Subtract

...

Symbolic manipulations

Compare two numbers

Substitute one string of letters and numbers for another

...

But when trillions of these simple operations are arranged in the right order, amazing computations can be carried out:

forecasting tomorrow's weather 

deciding where to drill for oil 

finding which places a person's most likely to visit 

figuring out who would make a great couple  

...





Article [Talk](#)

From Wikipedia, the free encyclopedia

For other uses, see [Computer science \(disambiguation\)](#).

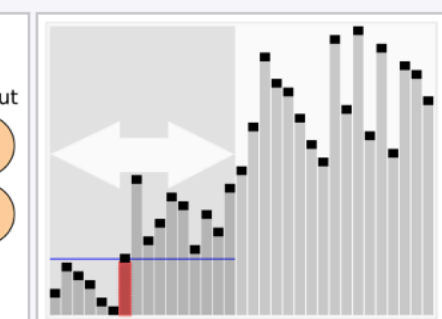
Computer science is the study of [computation](#), [information](#), and [automation](#).^{[1][2][3]}

Computer science spans **theoretical disciplines** (such as **algorithms**, **theory of computation**, and **information theory**) to **applied disciplines** (including the design and implementation of **hardware** and **software**).^{[4][5][6]}

Algorithms and [data structures](#) are central to computer science.^[7] The theory of computation concerns abstract [models of computation](#) and general classes of [problems](#) that can be solved using them. The fields of [cryptography](#) and [computer security](#) involve studying the means for secure communication and preventing [security vulnerabilities](#). [Computer graphics](#) and [computational geometry](#) address the generation of images. [Programming language theory](#) considers different ways to describe computational processes, and [database](#) theory concerns the management of repositories of data. [Human–computer interaction](#) investigates the interfaces through which humans and computers interact, and software engineering focuses on the design and principles behind developing software. Areas such as [operating systems](#), [networks](#) and [embedded systems](#) investigate the principles and design behind [complex systems](#). Computer architecture describes the construction of computer components and computer-operated equipment. [Artificial intelligence](#) and [machine learning](#) aim to synthesize goal-orientated processes such as problem-solving, decision-making, environmental adaptation, [planning](#) and learning found in humans and animals. Within artificial intelligence, [computer vision](#) aims to understand and process image and video data, while [natural language processing](#) aims to understand and process textual and linguistic

The logic diagram shows two inputs, A and B, connected to two gates. The top gate is an XOR gate, with its output labeled S. The bottom gate is an AND gate, with its output labeled C. The inputs A and B are connected to both gates: A to the top-left input of the XOR gate and the top-left input of the AND gate; B to the top-right input of the XOR gate and the bottom input of the AND gate.

Computer architecture



Computational complexity theory

- History
- Outline
- Glossary
- Category



Game 2 | Jonathan Gordon - Auto-Match Player (Black to Move)

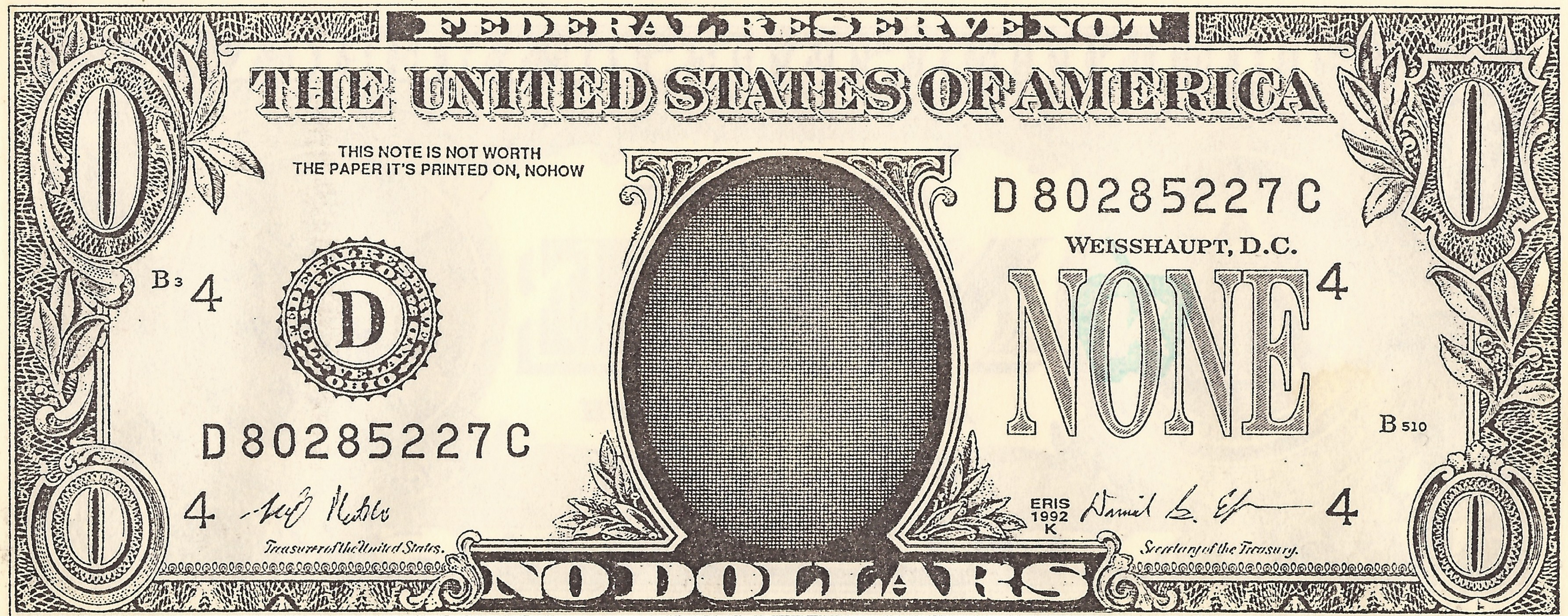


The magic of a computer is its ability to become almost anything you can imagine...

The magic of a computer is its ability to become almost anything you can imagine...

...as long as you can explain *exactly* what that is.

When we program a computer to do something,
everything needs to be described precisely.



eblong.com/zarf/zero-bill.html

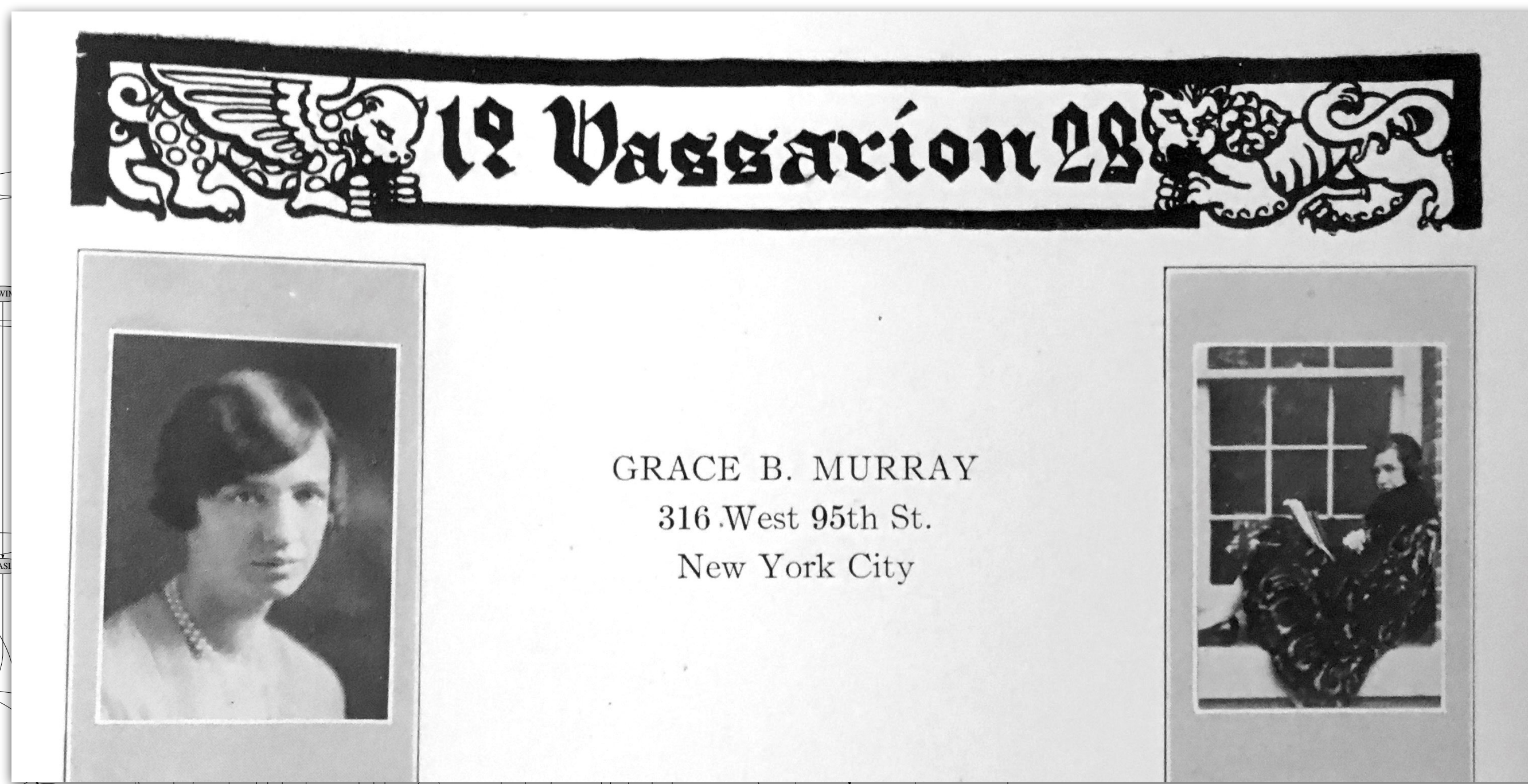
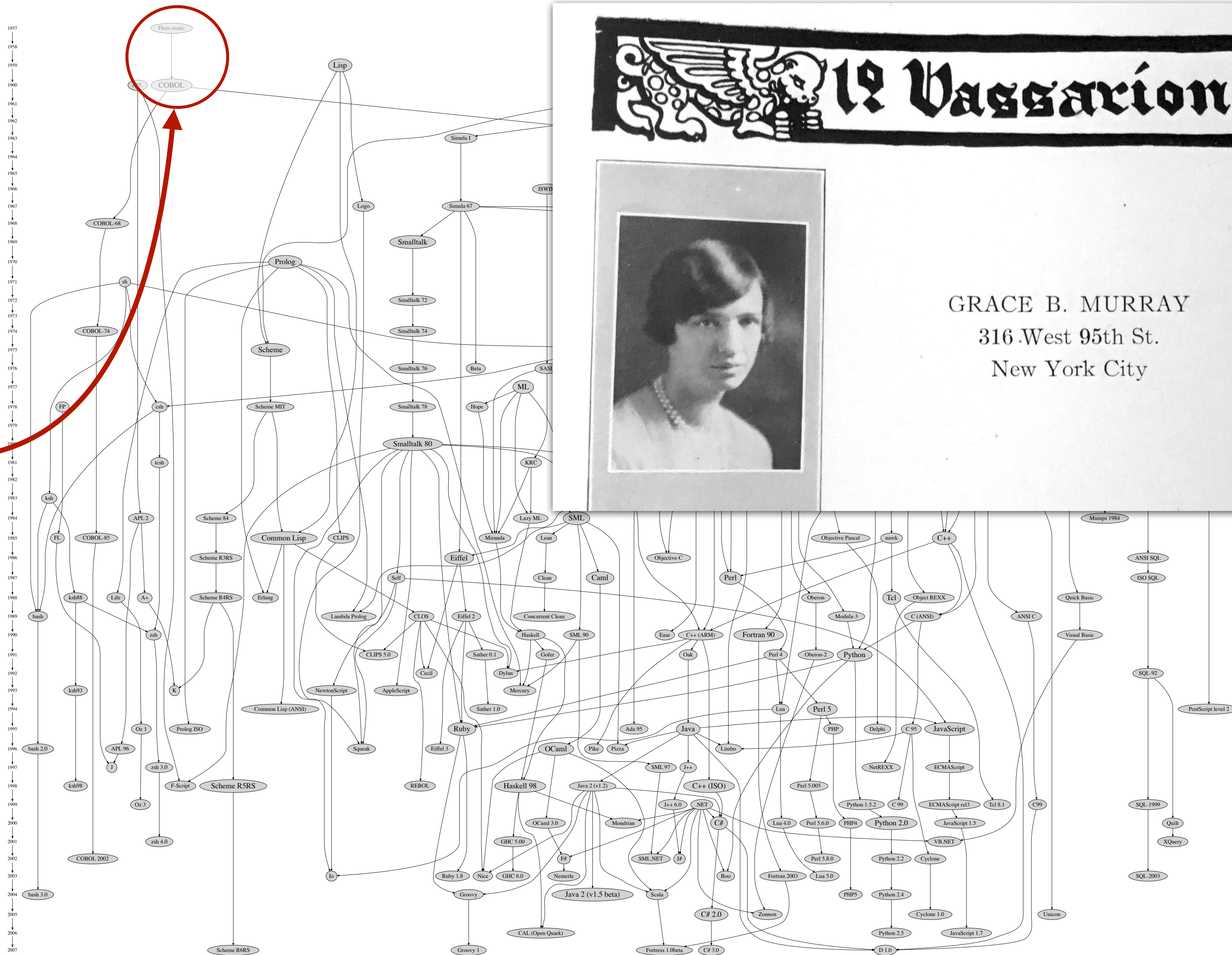
When computers behave intelligently, it's because a person used *their* intelligence to design an intelligent program.

To tell the computer exactly how to behave, we give it instructions using a *programming language*.

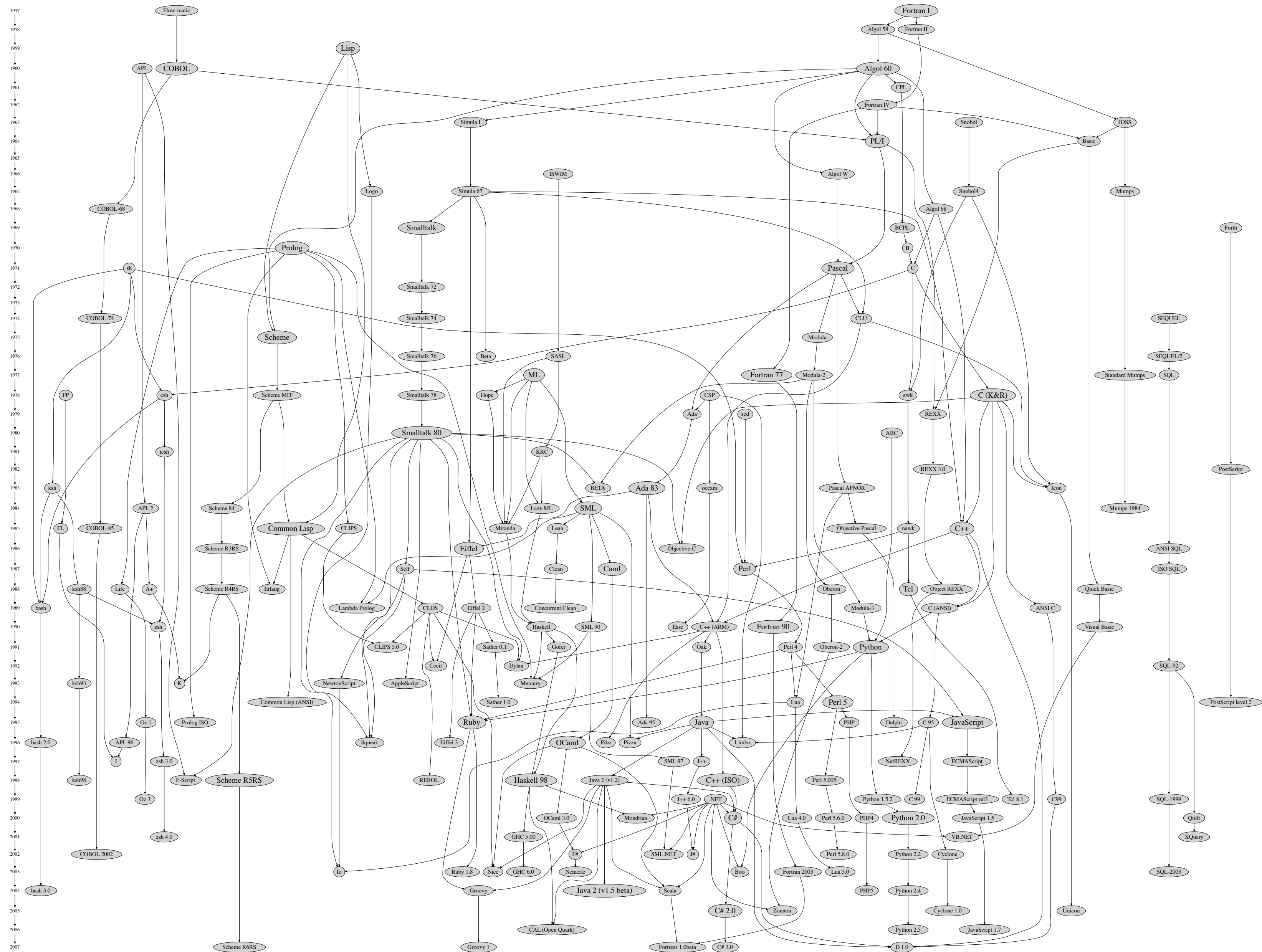




Grace Hopper



GRACE B. MURRAY
316 West 95th St.
New York City



github.com/stereobooster/programming-languages-genealogical-tree

There are many programming languages due to

intended use

history

habit

taste

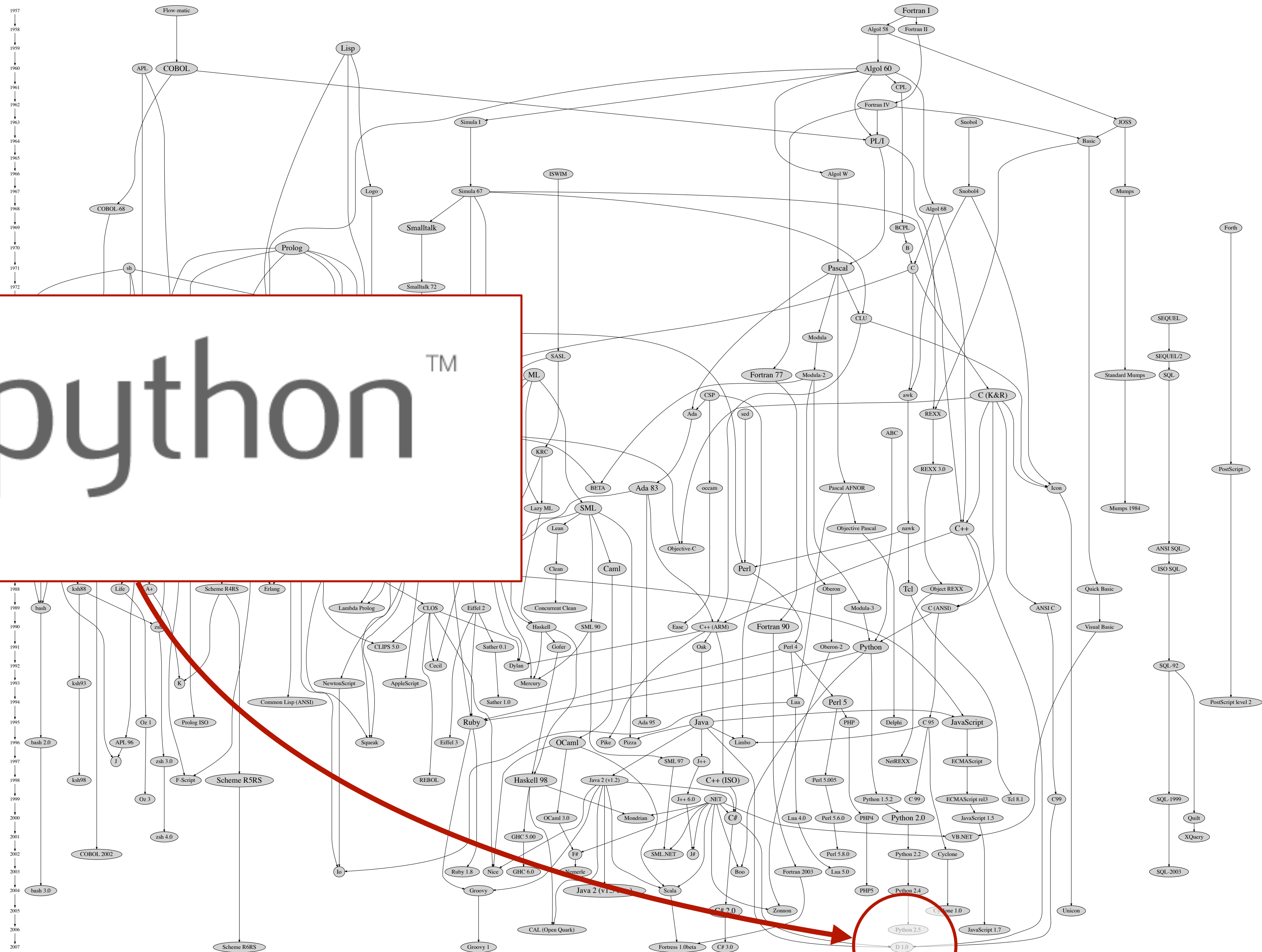
Ancient history (my childhood)



Ancient history (my childhood)

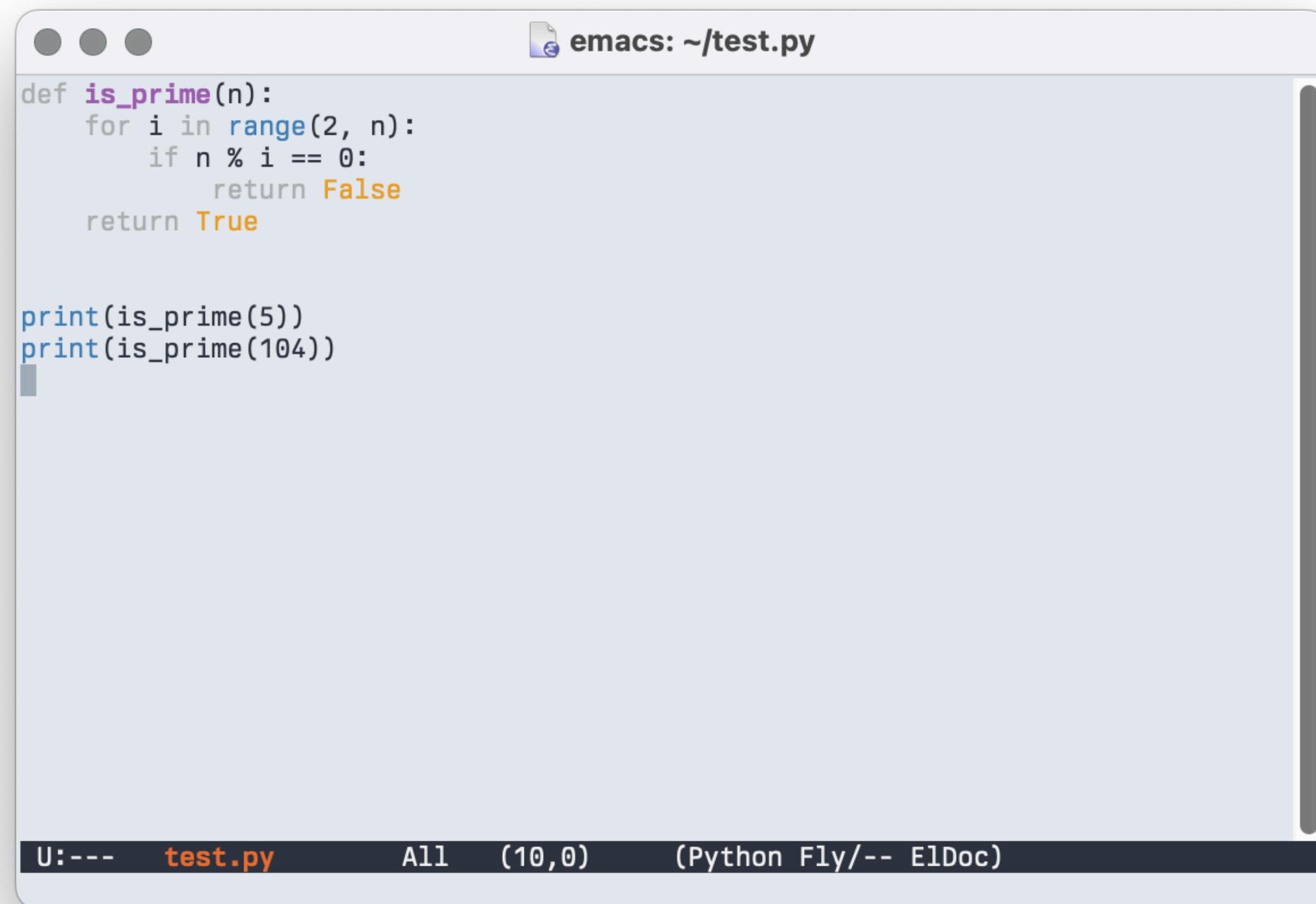


In this course, we'll be working in a slightly more modern programming language.



github.com/stereobooster/programming-languages-genealogical-tree

The traditional way of writing code is to use a text editor and then run the code in a command-line interface.

A screenshot of the Emacs text editor window. The title bar shows 'emacs: ~/test.py'. The editor contains Python code for a prime number checker. The status bar at the bottom shows 'U:--- test.py All (10,0) (Python Fly/-- ElDoc)'.

```
def is_prime(n):  
    for i in range(2, n):  
        if n % i == 0:  
            return False  
    return True  
  
print(is_prime(5))  
print(is_prime(104))
```

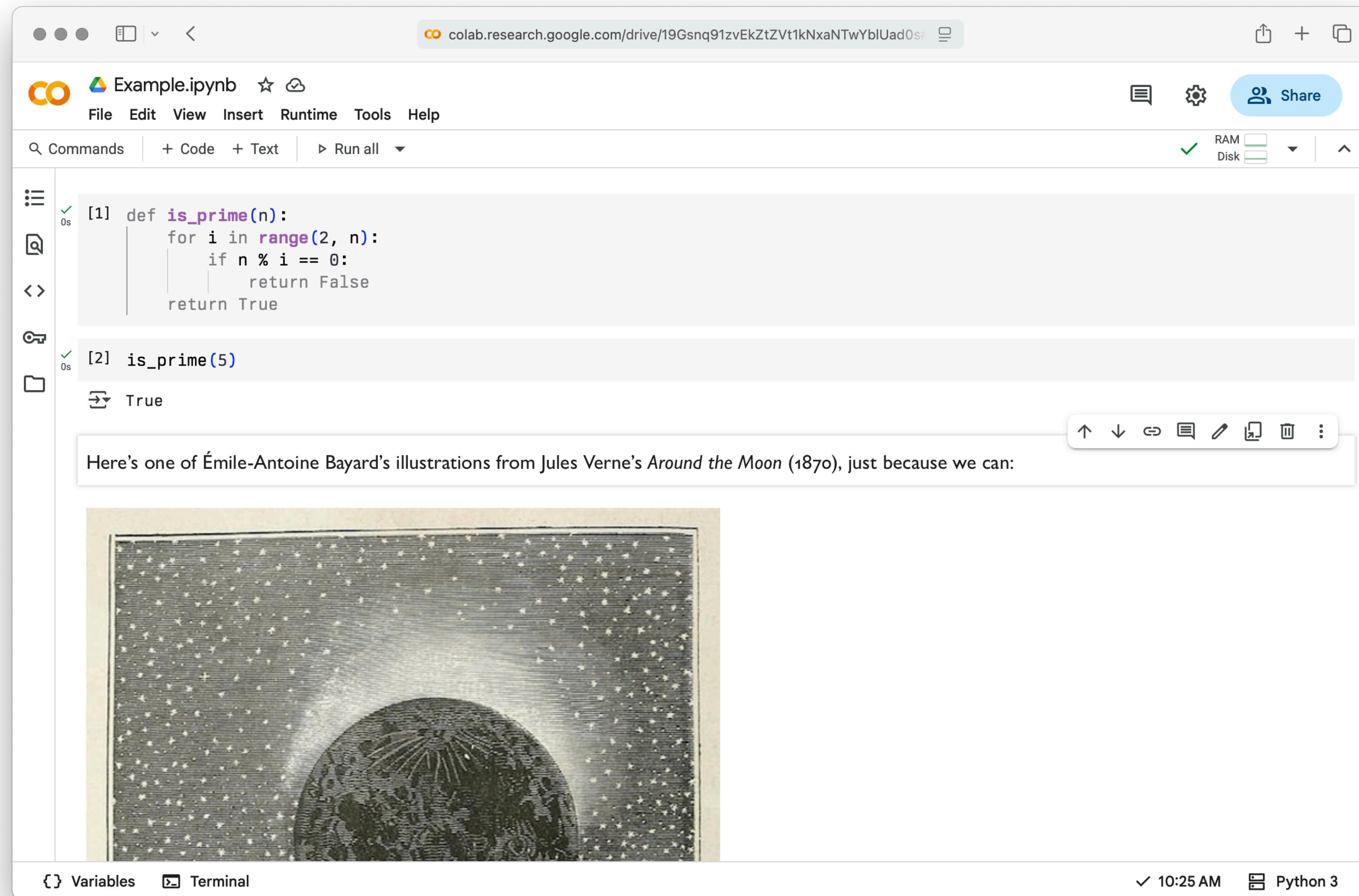
Emacs, a popular text editor

A screenshot of a terminal window. The title bar shows 'jgordon@jgordon: ~ [main]'. The terminal shows the command to run a Python script and its output.

```
; python3 test.py  
True  
False  
;
```

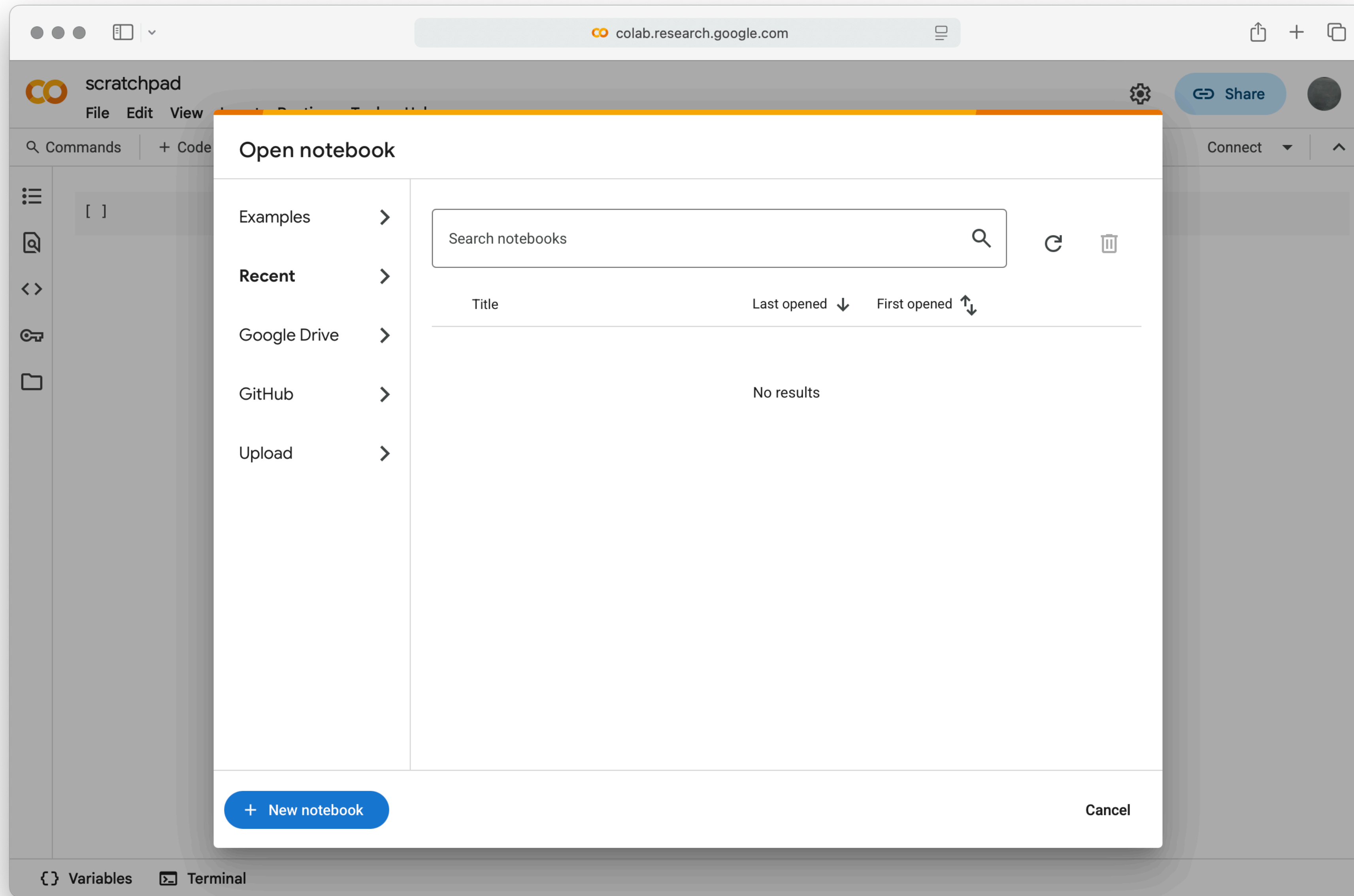
Command-line interface

While we could do everything in command-line interfaces, they aren't the best suited for data science work, which often requires *visualizations* and *written reports*.



Jupyter notebooks allow us to write and run code in a single document, together with accompanying text, tables, and images.

For this class, we'll use Colab, Google's version of Jupyter notebooks, which let you log in with your Vassar account and store the notebooks you write in your Google Drive.



colab.research.google.com

See *notebook* for example.

- 1 *Data design* Identify and organize the data needed to solve a problem
- 2 *Computational problem solving* Break a problem down into subproblems that can be solved with computations
- 3 *Programming* Express computations over the data
- 4 *Testing* Test those computations to make sure they're doing what they're supposed to

0 *Society*

Think about whether it's a good idea to solve the problem – and how your solution might affect the world around you.

You will leave this course with applicable skills that you can use *even if you don't take any future computer science or data science courses.*

Course information

docs.google.com/forms/d/e/1FAIpQLSdc7f

+



CMPU 100 §I Student information

Fall 2026

Please fill out this short form to help me better prepare for the start of the semester.

jgordon@vassar.edu [Switch account](#)

* Indicates required question

Email *

☐ Record jgordon@vassar.edu as the email to be included with my response

What name would you like me to call you? *

Your answer

Are you trying to add this class, currently enrolled, or planning to drop? *

☐ I would like to add this class.

☐ I am currently enrolled in this class.

forms.gle/VgqTc9kGvbGTSW6C6

<i>Class</i>	Monday	1:30– 2:45 p.m.
	Wednesday	1:30– 2:45 p.m.
<i>Lab</i>	Friday	1:00– 3:00 p.m.

Sanders Classroom 006

cs.vassar.edu/~cs100/

CMPU 100

LABS

ASSIGNMENTS

RESOURCES

GRADESCOPE

Programming with Data

Fall 2026 · §1

Monday1:30–2:45 p.m.

Wednesday1:30–2:45 p.m.

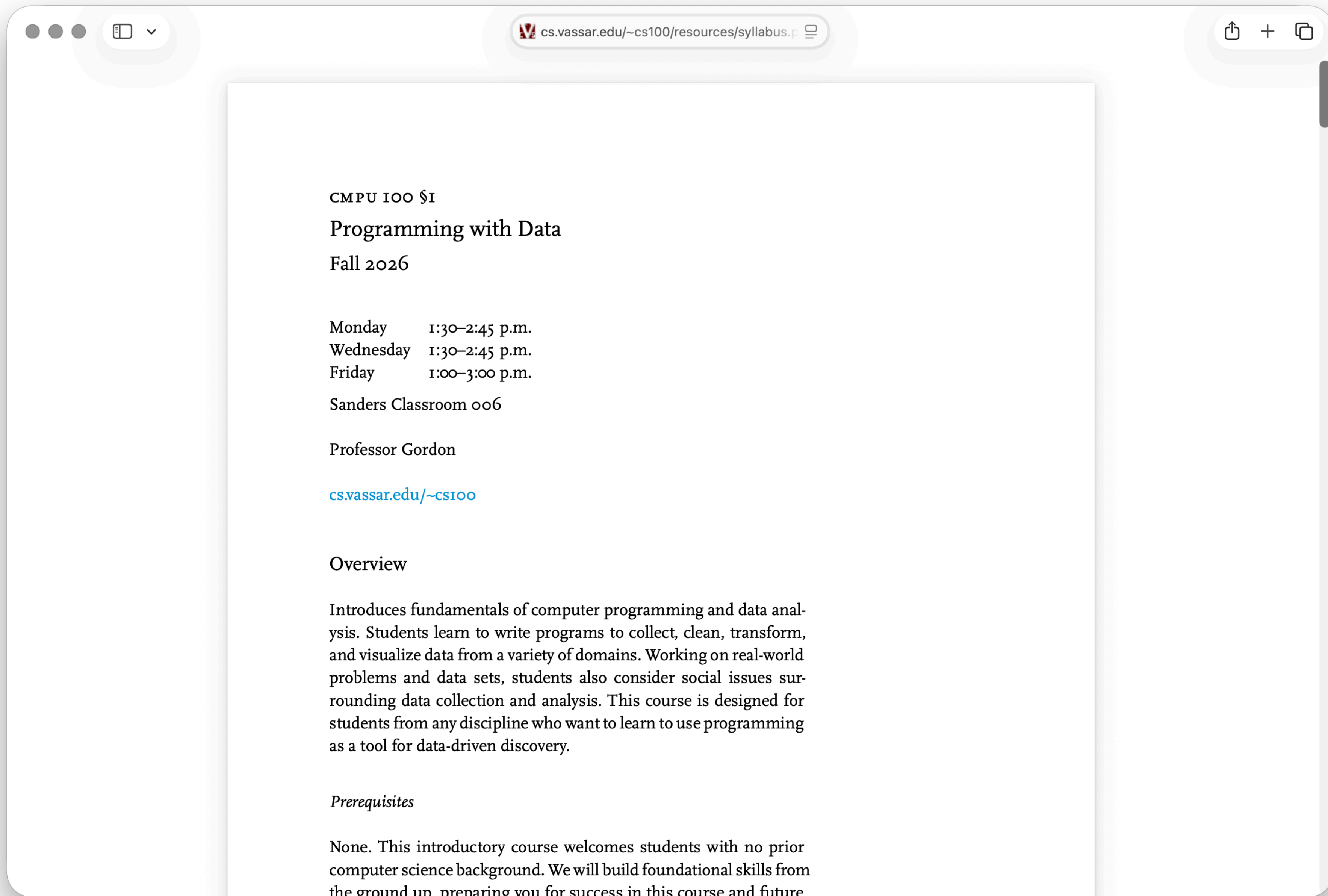
Friday1:00–3:00 p.m.

Sanders Classroom 006

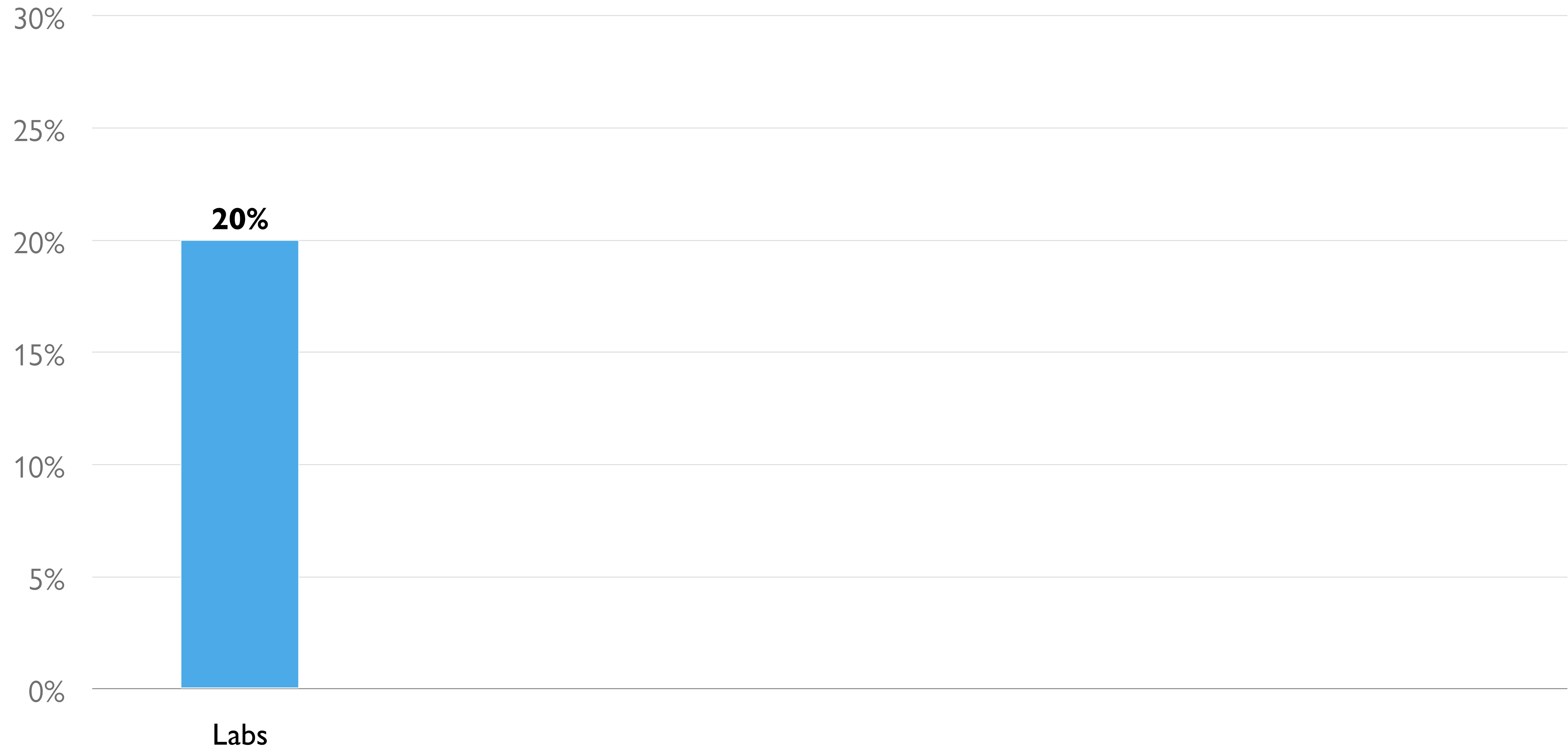
[Professor Gordon](#)

↓ Current

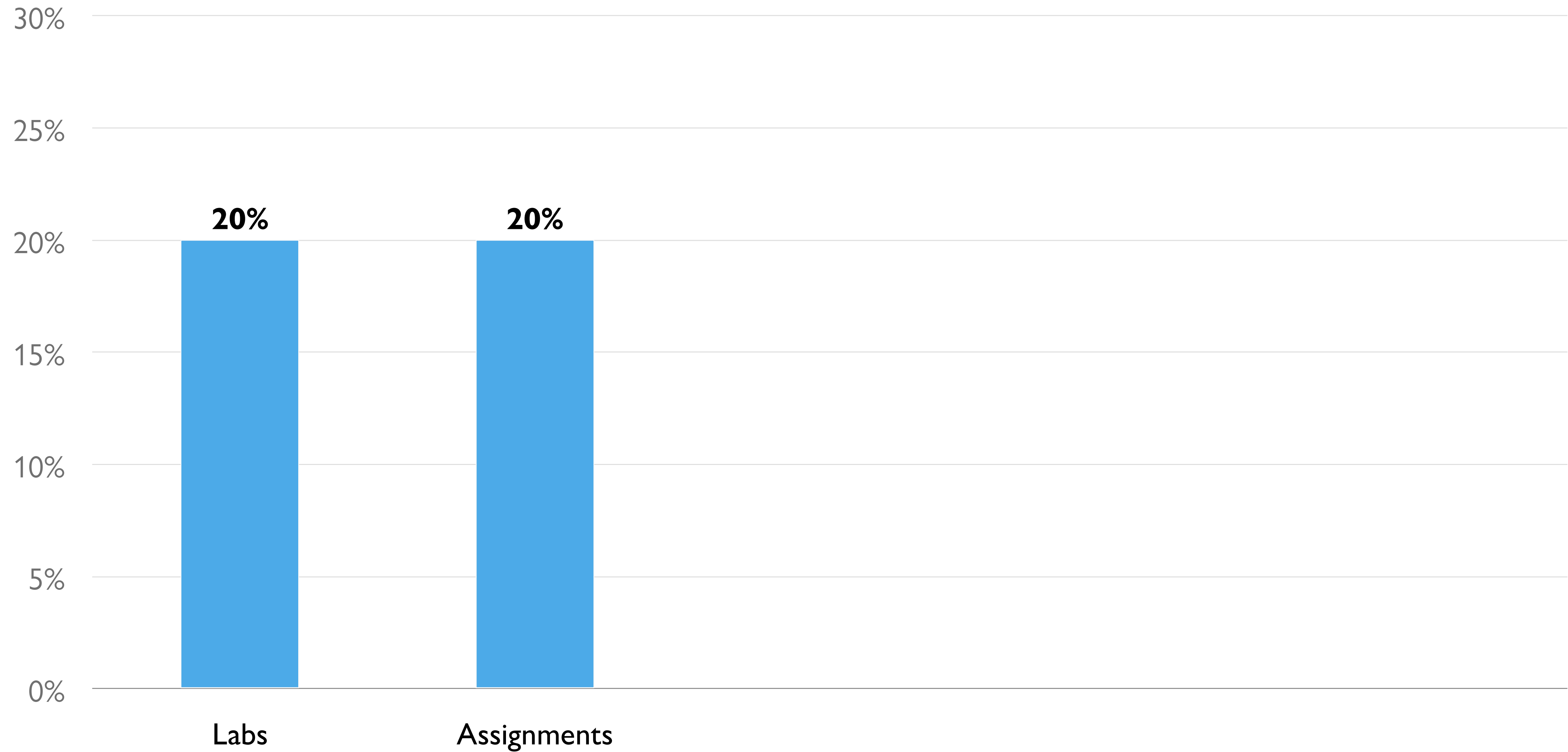
Part 1: Foundations	Monday	Wednesday	Friday
<i>Introduction</i>	<i>Aug. 31</i>	<i>Sep. 2</i>	<i>Sep. 4</i>
– Read Syllabus – Read CIT §3.1–3.3 and CIT §4.1–4.2	Class 1 Programming with data	Class 2 Expressions, values, and	Lab 0 Getting started



Grading



Grading





<≡

CMPU 100 §1

Programming with Data

Dashboard

Regrade Requests

Instructor

J. Gordon

Course Actions

Unenroll From Course

Account

^

www.gradescope.com/courses/1379400

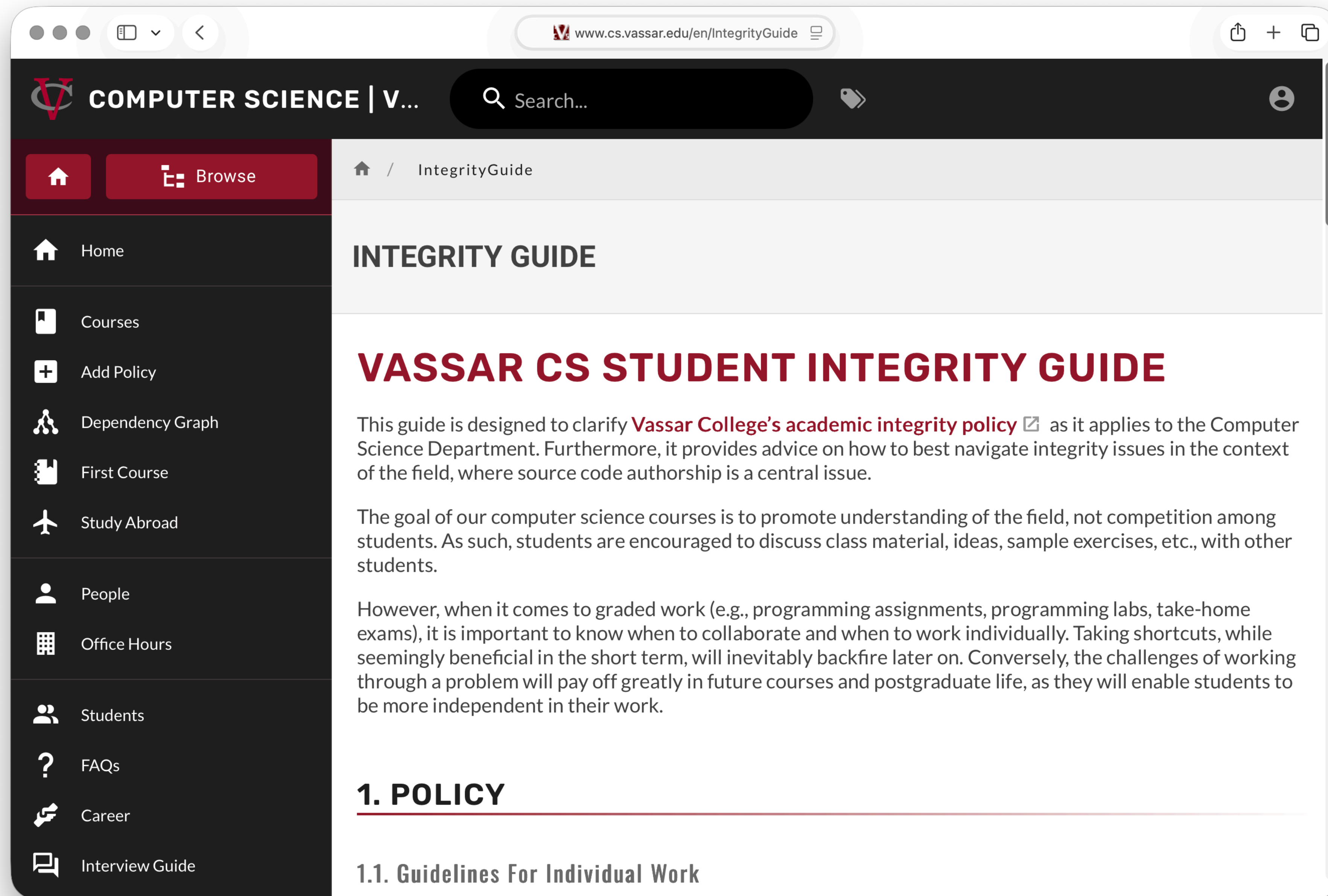
↑+□

CMPU 100 §1

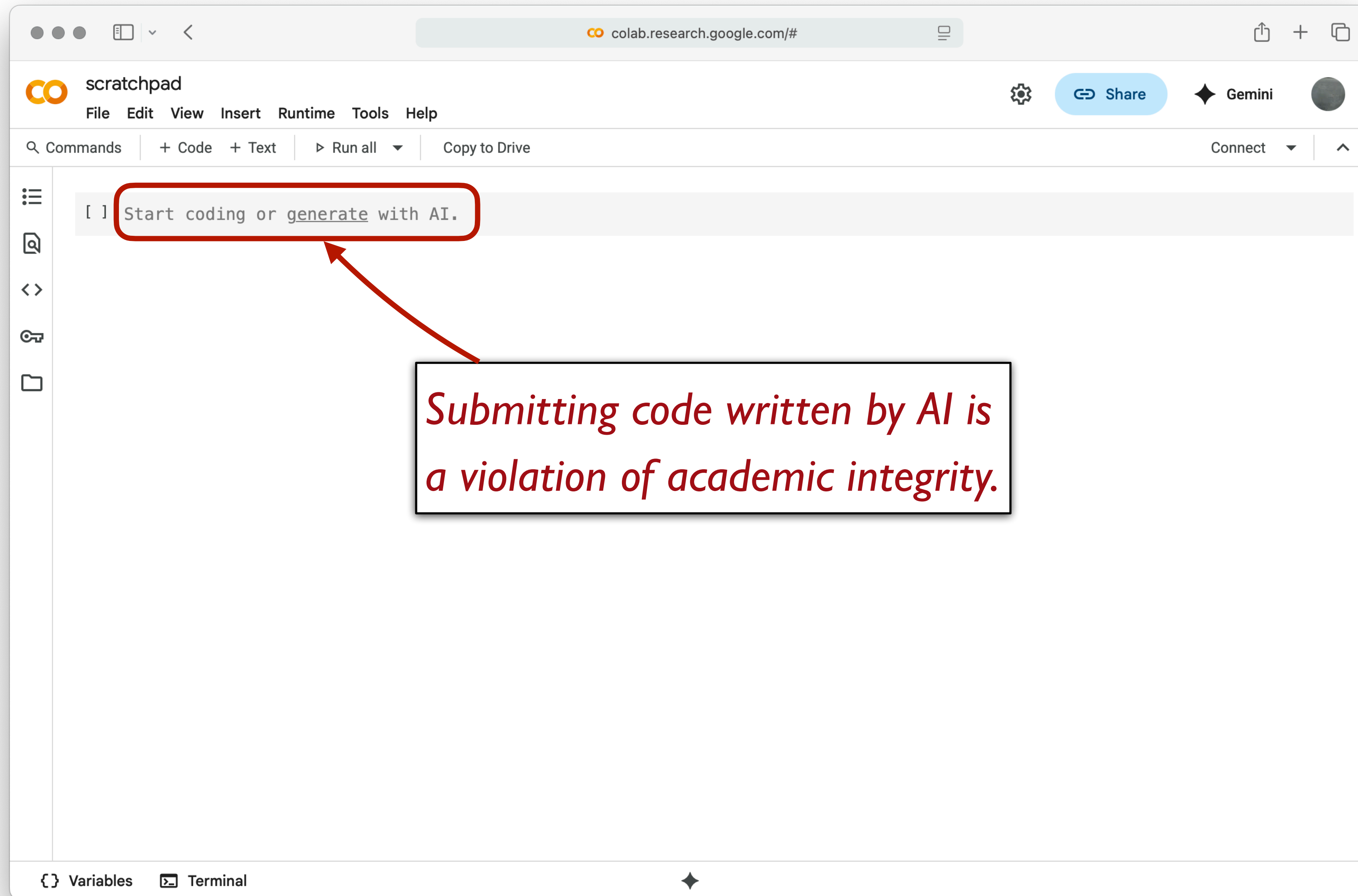
Fall 2026

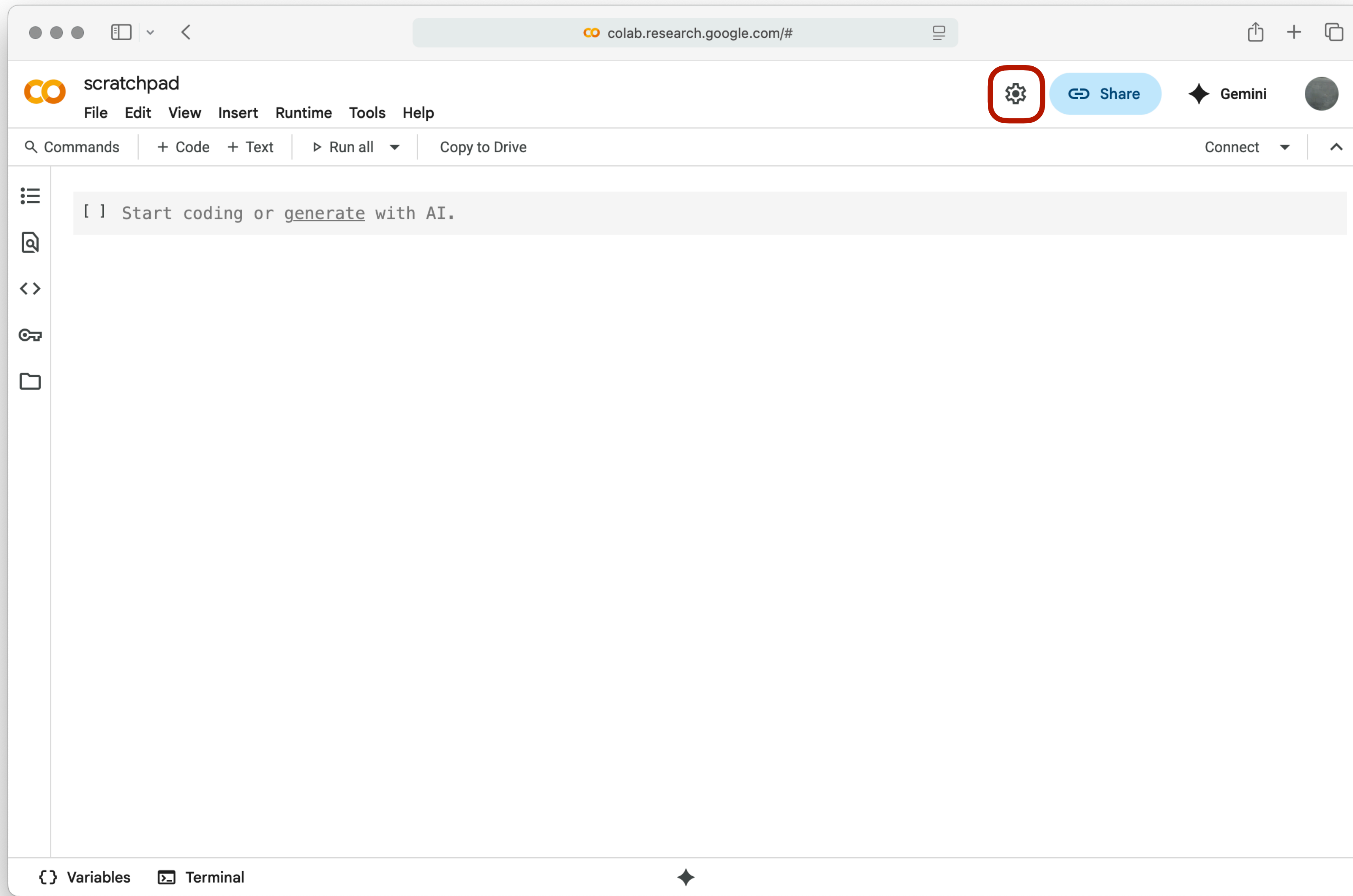
Course ID: 1379400

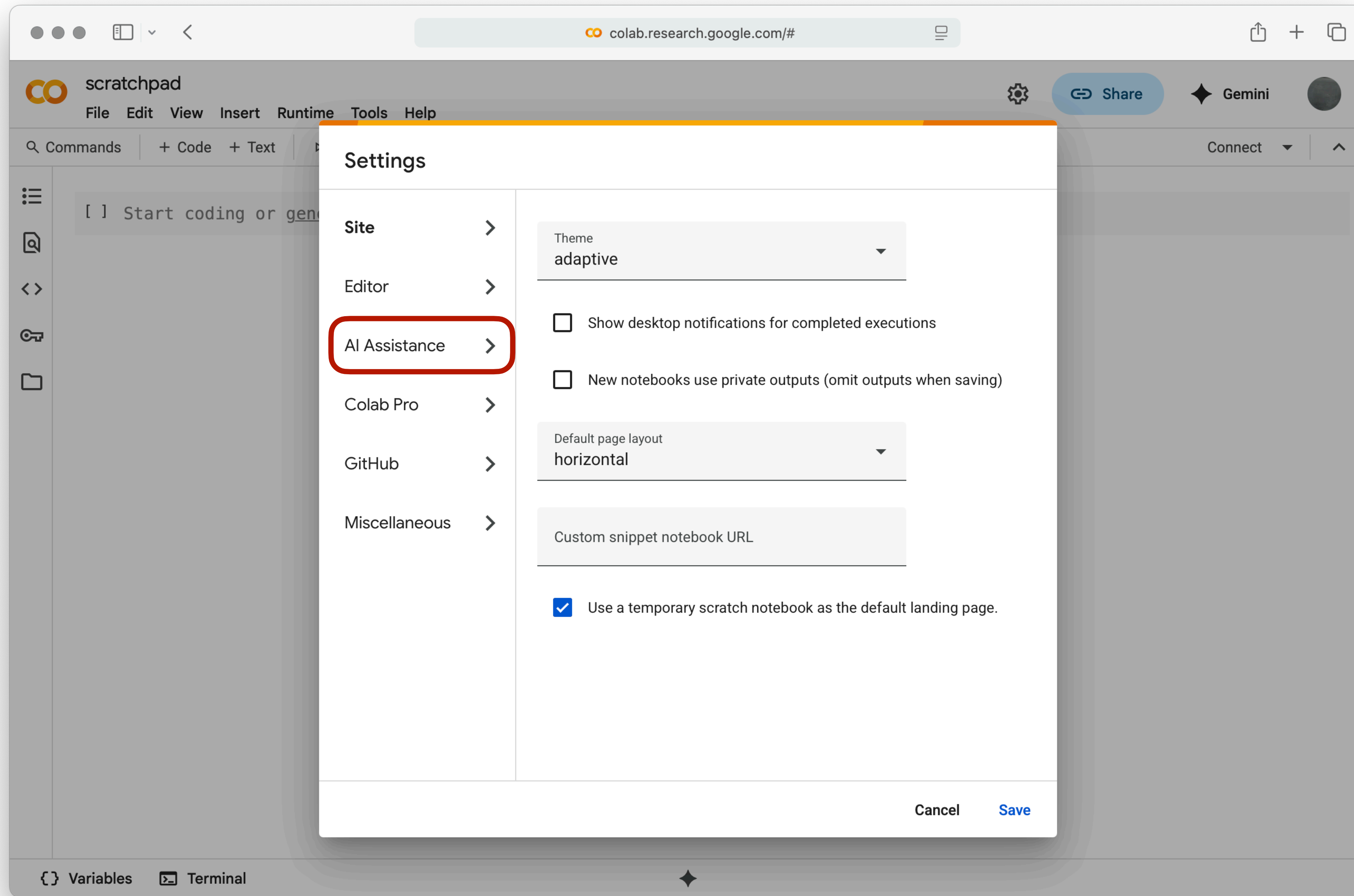
Name	Status	Released	Due (EDT)
Your instructor hasn't released any assignments yet.			

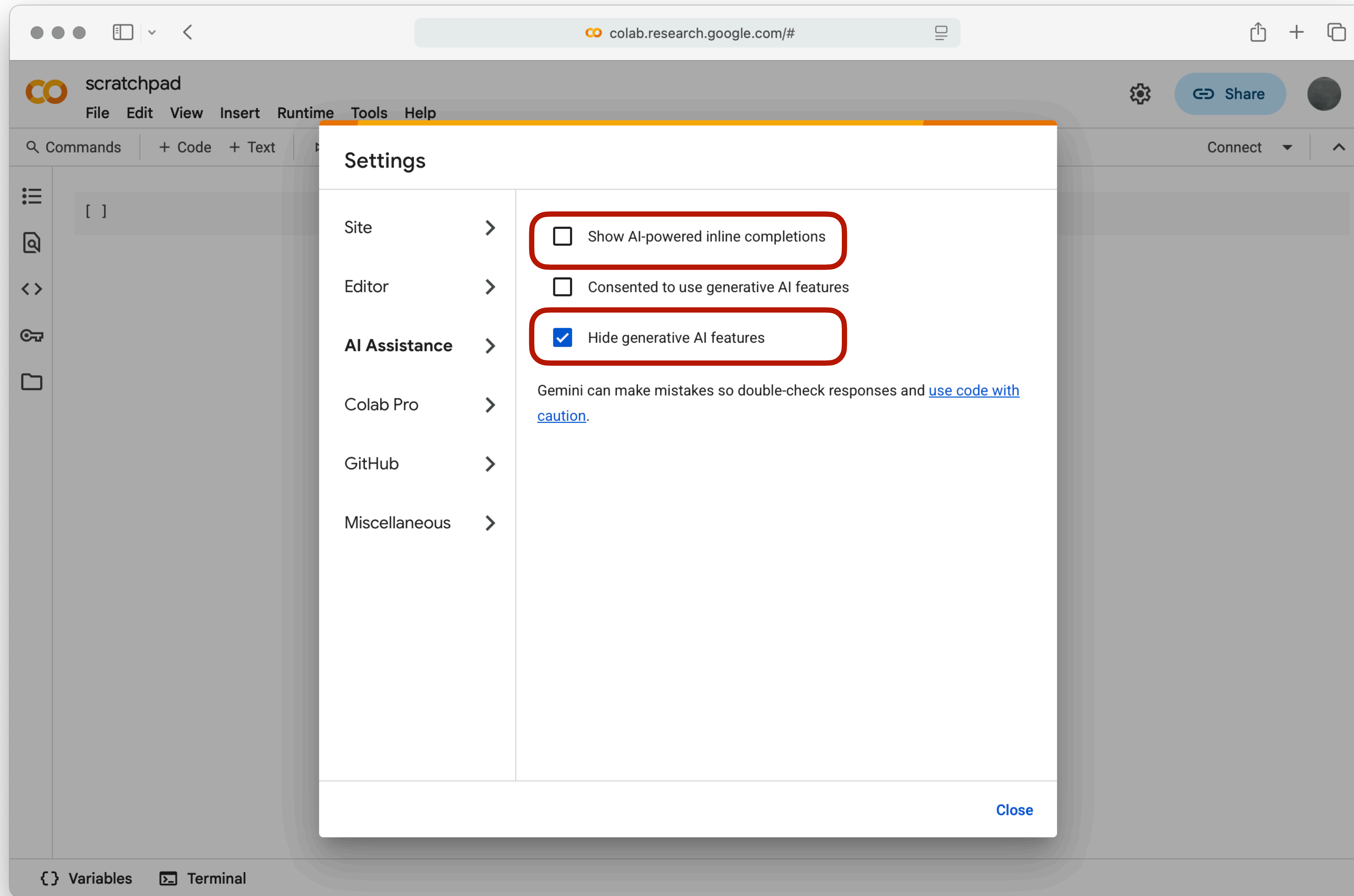


cs.vassar.edu/en/IntegrityGuide

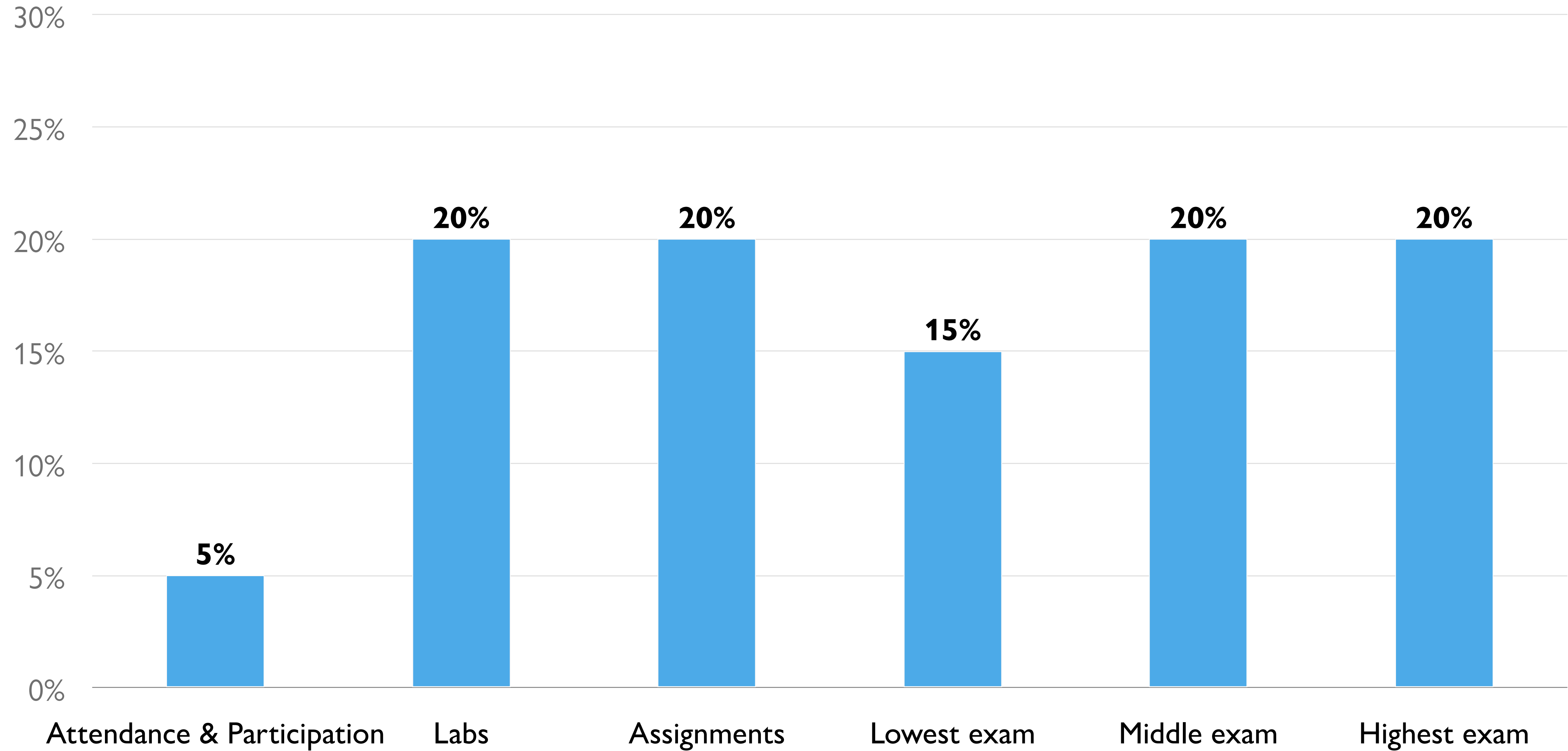








Grading





“All through our education, we are being taught a kind of reverse mindfulness. A kind of Future Studies where – via the guise of mathematics, or literature, or history, or computer programming, or French – we are being taught to think of a time different to the time we are in. Exam time. Job time. When-we-are-grown-up time.

“To see the act of learning as something not for its own sake but because of what it will get you reduces the wonder of humanity. We are thinking, feeling, art-making, knowledge-hungry, marvelous animals, who understand ourselves and our world through the act of learning. It is an end in itself. It has far more to offer than the things it lets us write on application forms. It is a way to love living right now.”

Matt Haig, *Notes on a Nervous Planet*

We've got a big journey ahead of us. I hope you're excited!

