Preliminaries

Science

Computer Science

CS 331: Data Structures and Algorithms

Michael Lee <lee@iit.edu>

ILLINOIS INSTITUTE OF TECHNOLOGY
College of Science

Michael Lee

- lee@iit.edu
- http://moss.cs.iit.edu
- Office hours by appointment (Tue/Thu 11AM-1PM)

Agenda

- Course overview & Administrivia
  - Prerequisites
  - Topics & Resources
  - Grading
  - Dev environment & Class procedures

Data Structures

- How do we store, organize, and retrieve data on a computer?

& Algorithms

- How can we efficiently (in space/time) carry out some typical data processing operations?
- How do we analyze and describe their performance?

Prerequisites

- I assume you are …
  - fluent in some programming language
    - familiar with procedural & OO paradigms
  - comfortable with development processes:
    - compilation, debugging, testing

Python

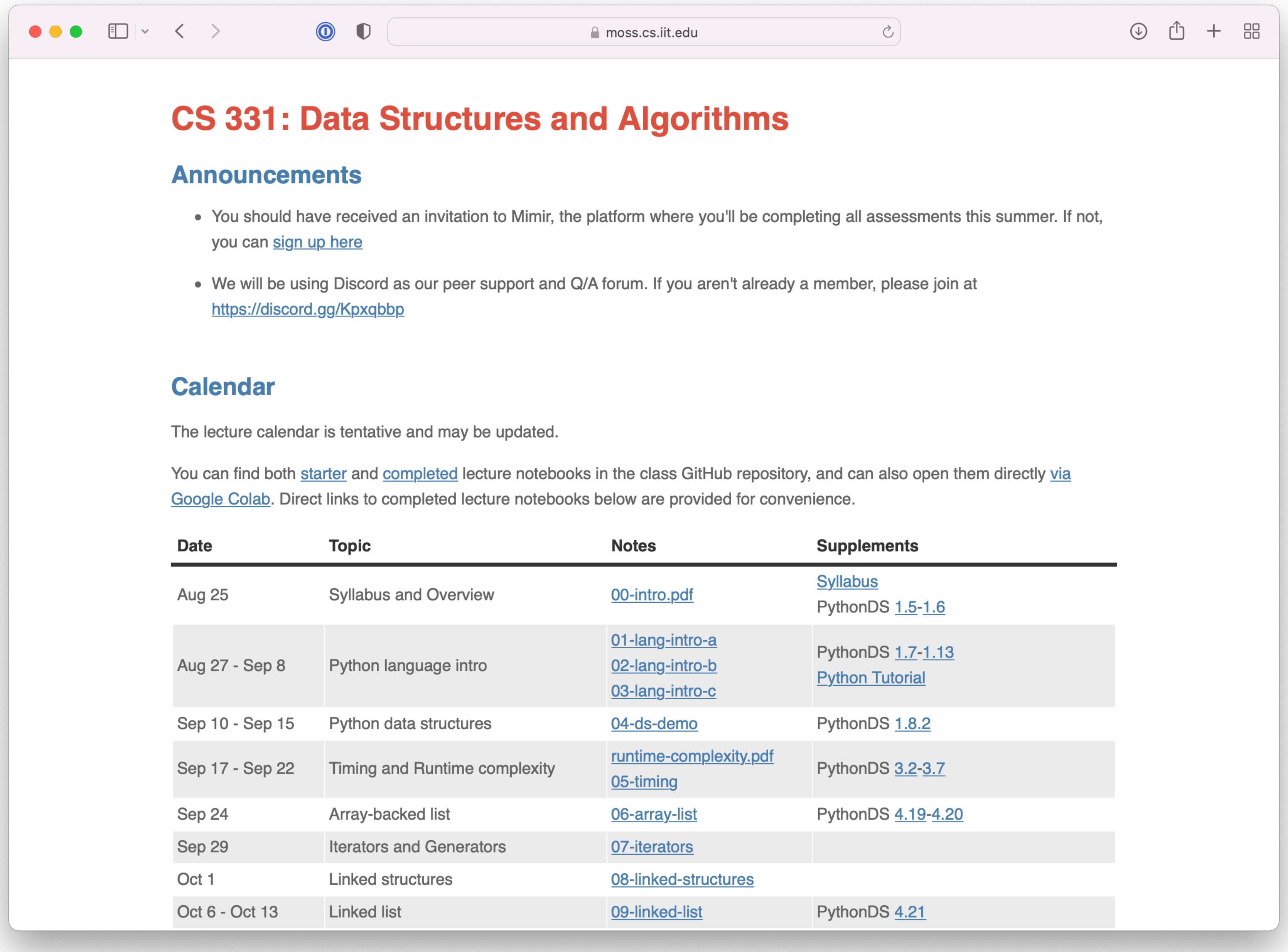
- We’ll use the Python programming language to explore data structures & algorithms
- Easy-to-learn, clean (“one obvious way to do” things), and popular language
  - Ton of useful, powerful libraries

Topics

- Python crash course
- Algorithmic analysis
- Linear data structures (Lists, Stacks, Queues)
- Mapping structures (Hashtables and Trees)
- Recursion

Class Resources

1. Course website: moss.cs.iit.edu/cs331
    - static information
        - lecture calendar, slides, links to external resources, etc.



CS 331: Data Structures and Algorithms

Announcements

- You should have received an invitation to Mimir, the platform where you'll be completing all assessments this summer. If not, you can sign up here
- We will be using Discord as our peer support and Q/A forum. If you aren't already a member, please join at https://discord.gg/Kpxqbbp

Calendar

The lecture calendar is tentative and may be updated.

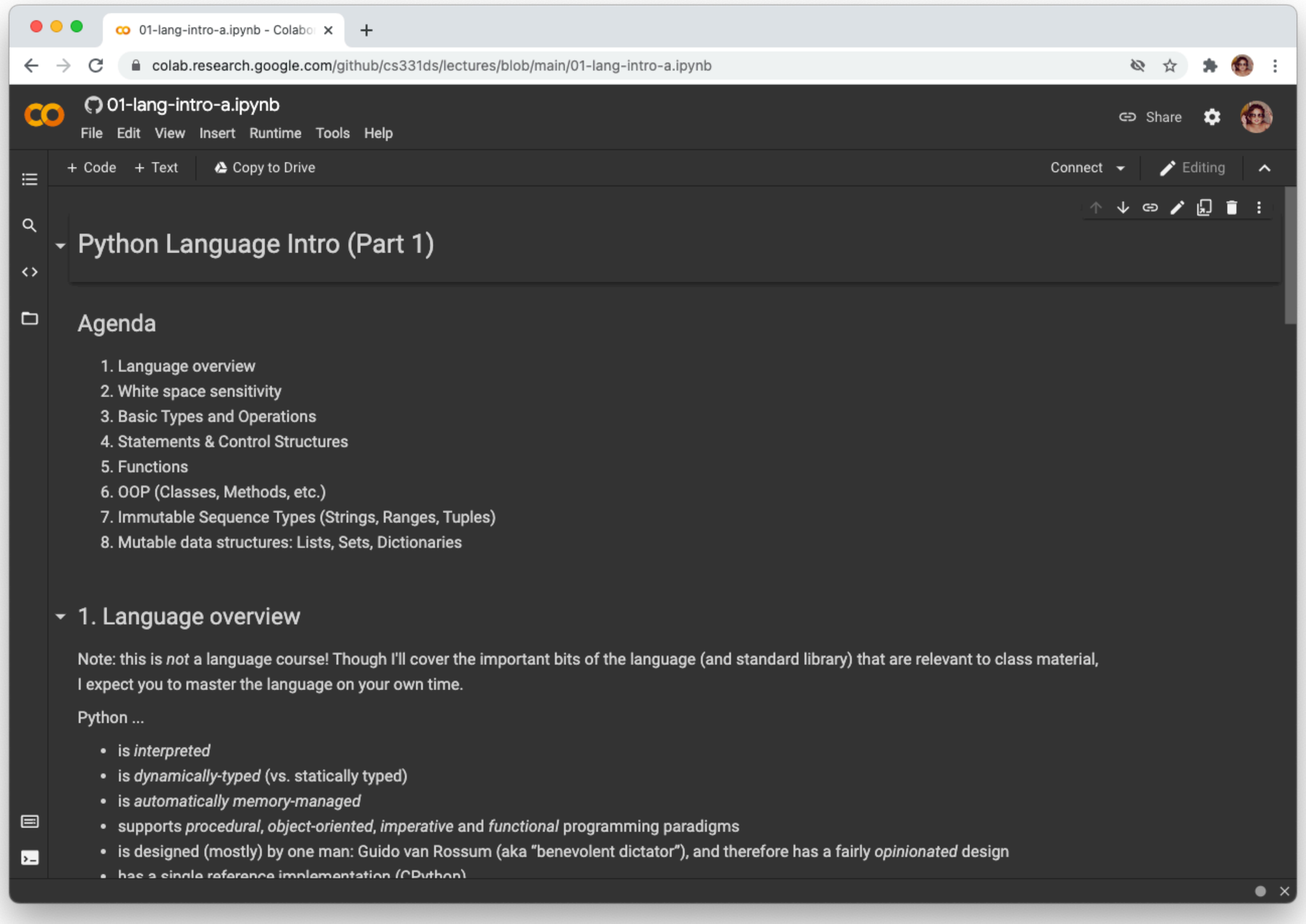
You can find both starter and completed lecture notebooks in the class GitHub repository, and can also open them directly via Google Colab. Direct links to completed lecture notebooks below are provided for convenience.

Date	Topic	Notes	Supplements
Aug 25	Syllabus and Overview	00-intro.pdf	Syllabus PythonDS 1.5-1.6
Aug 27 - Sep 8	Python language intro	01-lang-intro-a 02-lang-intro-b 03-lang-intro-c	PythonDS 1.7-1.13 Python Tutorial
Sep 10 - Sep 15	Python data structures	04-ds-demo	PythonDS 1.8.2
Sep 17 - Sep 22	Timing and Runtime complexity	runtime-complexity.pdf 05-timing	PythonDS 3.2-3.7
Sep 24	Array-backed list	06-array-list	PythonDS 4.19-4.20
Sep 29	Iterators and Generators	07-iterators	
Oct 1	Linked structures	08-linked-structures	
Oct 6 - Oct 13	Linked list	09-linked-list	PythonDS 4.21

Class Resources

2. Google Colaboratory

   - interactive lecture notebooks


01-lang-intro-a.ipynb - Colabo
colab.research.google.com/github/cs331ds/lectures/blob/main/01-lang-intro-a.ipynb
01-lang-intro-a.ipynb
File
Edit
View
Insert
Runtime
Tools
Help
Share
+ Code
+ Text
Copy to Drive
Connect
Editing

Python Language Intro (Part 1)

Agenda

1. Language overview
2. White space sensitivity
3. Basic Types and Operations
4. Statements & Control Structures
5. Functions
6. OOP (Classes, Methods, etc.)
7. Immutable Sequence Types (Strings, Ranges, Tuples)
8. Mutable data structures: Lists, Sets, Dictionaries

1. Language overview

Note: this is *not* a language course! Though I'll cover the important bits of the language (and standard library) that are relevant to class material, I expect you to master the language on your own time.

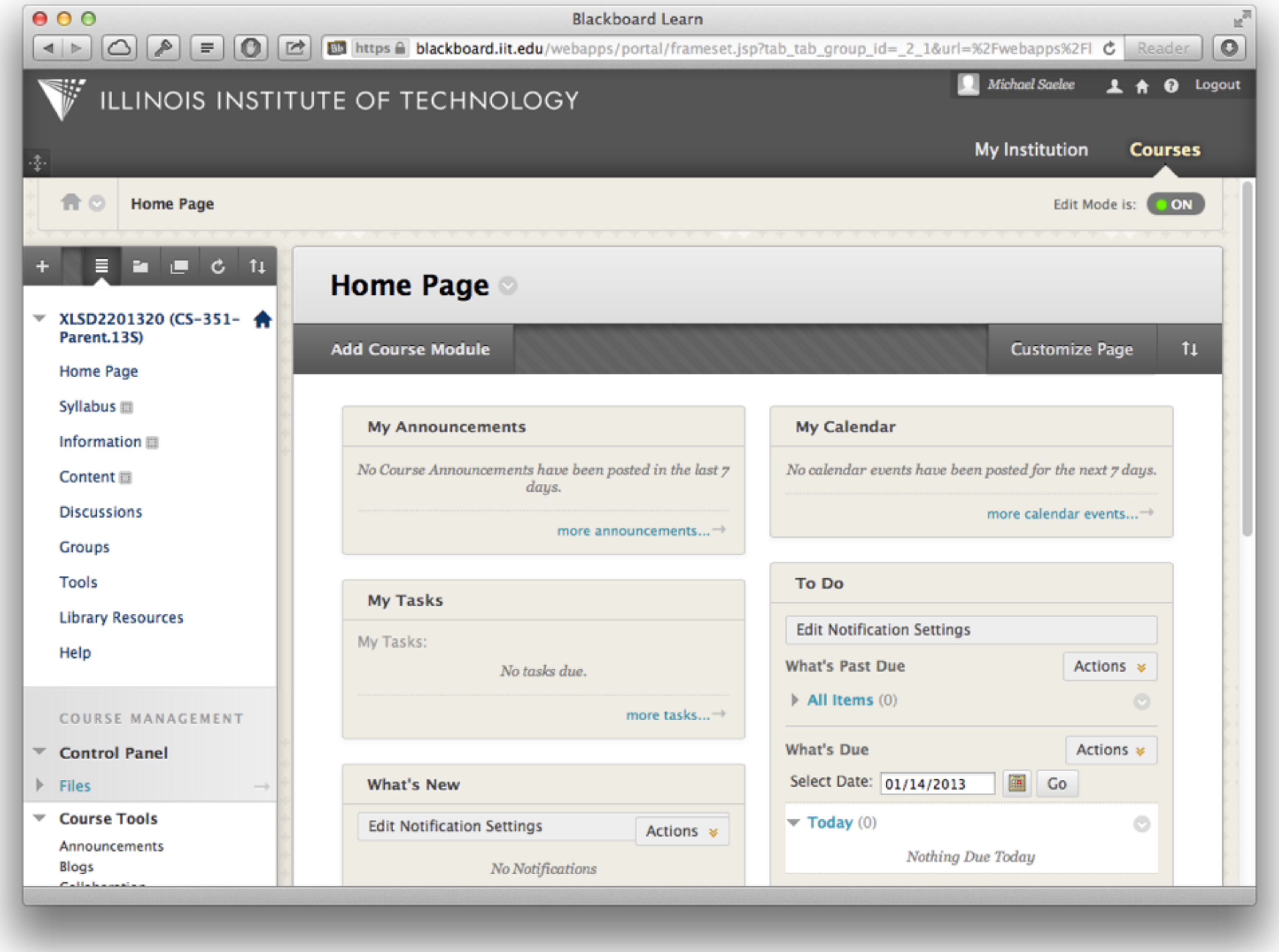
Python ...

- is *interpreted*
- is *dynamically-typed* (vs. statically typed)
- is *automatically memory-managed*
- supports *procedural*, *object-oriented*, *imperative* and *functional* programming paradigms
- is designed (mostly) by one man: Guido van Rossum (aka “benevolent dictator”), and therefore has a fairly *opinionated* design
- has a single reference implementation (CPython)

Class Resources

3. Blackboard

   - Final gradebook



No Course Announcements have been posted in the last 7 days.

No calendar events have been posted for the next 7 days.

My Tasks:

No tasks due.

Select Date: 01/14/2013

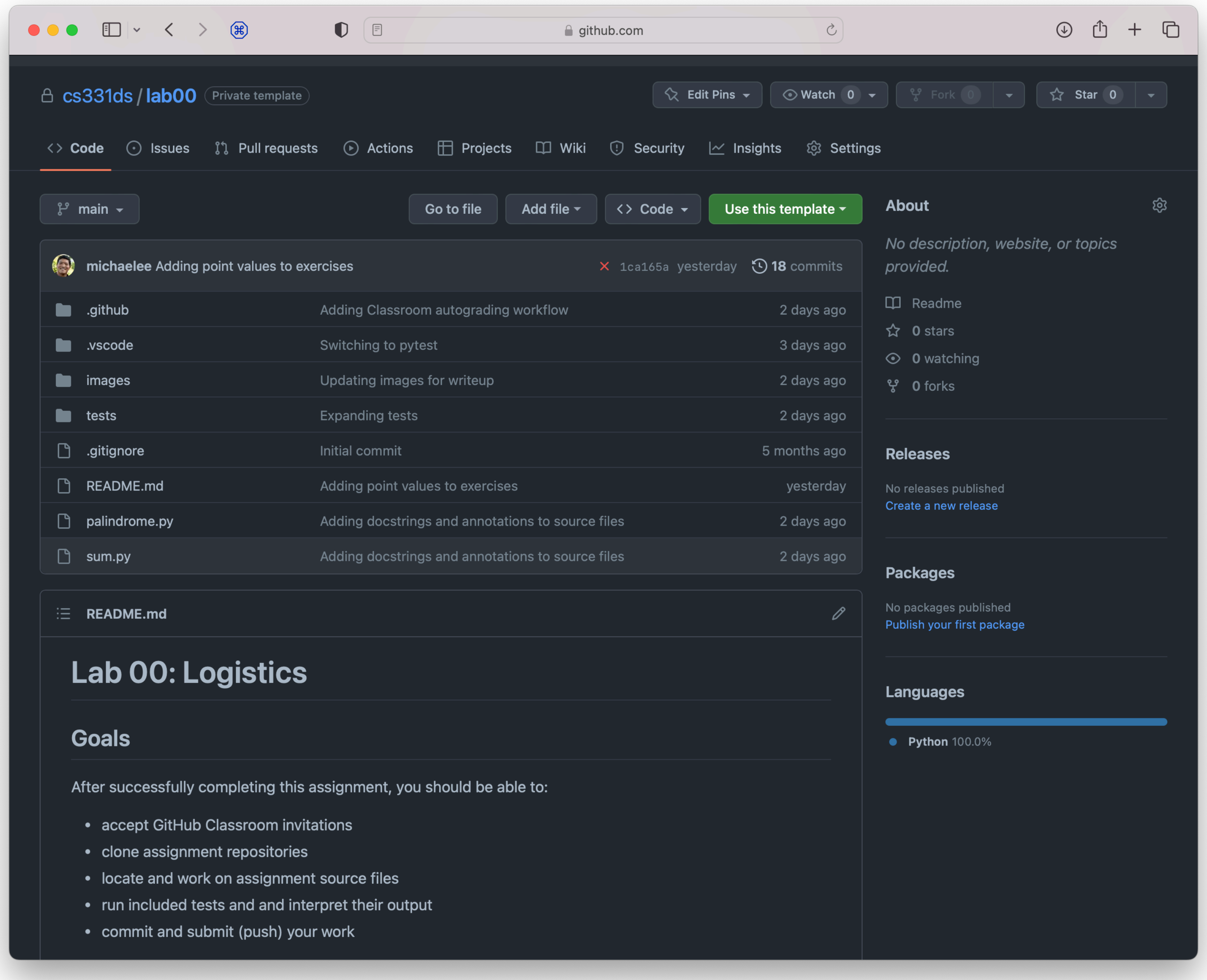
Nothing Due Today

No Notifications

Class Resources

4. GitHub

- Assignment distribution & submission



michaelee Adding point values to exercises ✕ 1ca165a yesterday 18 commits

.github	Adding Classroom autograding workflow	2 days ago
.vscode	Switching to pytest	3 days ago
images	Updating images for writeup	2 days ago
tests	Expanding tests	2 days ago
.gitignore	Initial commit	5 months ago
README.md	Adding point values to exercises	yesterday
palindrome.py	Adding docstrings and annotations to source files	2 days ago
sum.py	Adding docstrings and annotations to source files	2 days ago

README.md

Lab 00: Logistics

Goals

After successfully completing this assignment, you should be able to:

- accept GitHub Classroom invitations
- clone assignment repositories
- locate and work on assignment source files
- run included tests and and interpret their output
- commit and submit (push) your work

About

No description, website, or topics provided.

Readme
0 stars
0 watching
0 forks

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

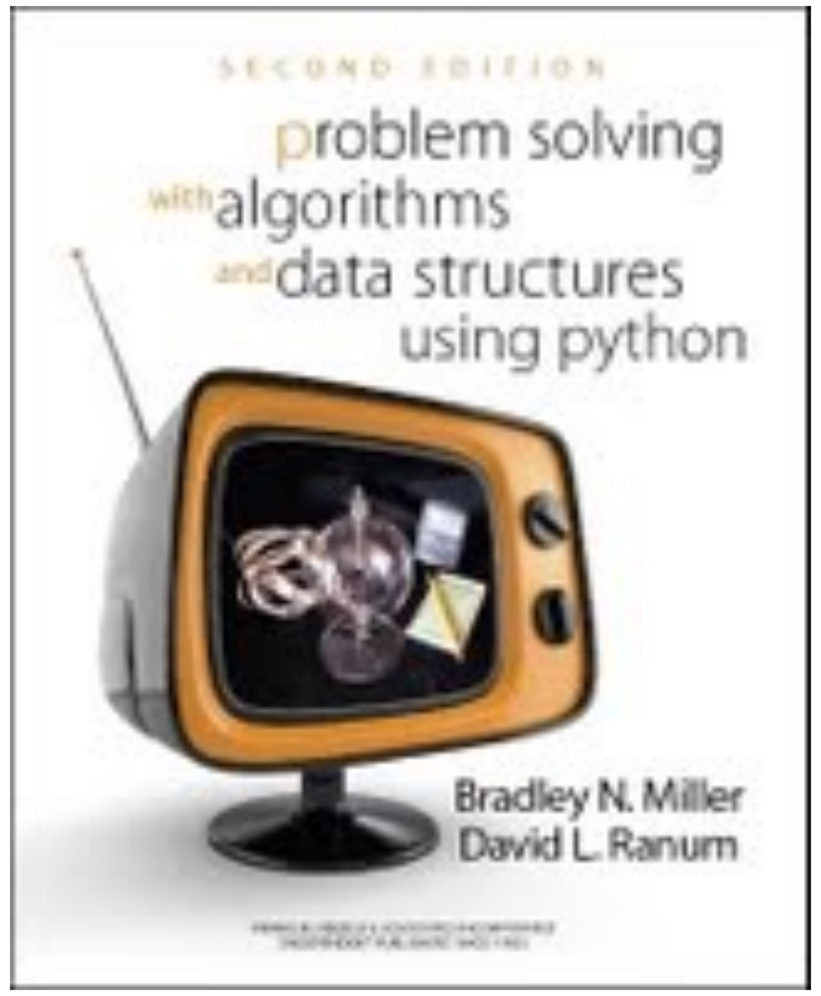
Languages

Python 100.0%

ILLINOIS INSTITUTE OF TECHNOLOGY
College of Science

Supplements

- The Python Tutorial (docs.python.org/3/)
- Problem Solving with Algorithms and Data Structures Using Python



Grading

- 40% Programming Assignments
- 60% Exams (3 total: 2 midterms + final)

Programming Assignments

- ~10 assignments
- All assignments are retrieved and submitted via GitHub
    - Provided codebase typically covered in preceding lectures
- Late policy: elastic over summer
    - Hard due date for everything: June 24

On Exams

- Tentative dates are on course website (**June 3**, **June 10**)
- All exams are cumulative (but will focus on recent material)
- Will be administered online, to be taken on your own computer (or in lab)
- Later exam scores, if higher, will improve earlier ones

```
>> scores = [60, 80, 75]
>> [max(scores[i:]) for i in range(3)]
[80, 80, 75]

>> scores = [75, 80, 100]
>> [max(scores[i:]) for i in range(3)]
[100, 100, 100]
```

Jupyter Notebook

- In-browser Python development platform
  - “Cells” can contain plain text, code, output (and more)
  - All lecture notes will be distributed as notebook files

Jupyter Notebook

- We strongly recommend installing Microsoft Visual Studio Code (VSCode) as an IDE for both lectures & labs
    - Install the Jupyter and Python extensions

Interactive Lectures

- Lecture notebooks available in course repository
  - Open on Google Colab or VSCode
    - Class is usually one long interactive demo
- Starter and completed lecture notebooks are available in the course GitHub repository