

Gateway Python Course Overview

Dr. Kai Presler-Marshall (“Dr. Kai”)

GETTING ACQUAINTED WITH GATEWAY PYTHON

Course Overview

- Who?
- Where?
- When?
- What?
- Why?
- How?

Course Overview: Who?

- Professor: Dr. Kai
 - I'm not really that picky on what you call me, as long as it's polite
 - Professor Presler-Marshall is fine too, just verbose 😊
- Office is in Malone 337
 - Feel free to drop by any time the door is open
- Email: kai@cs.jhu.edu



Course Overview: Who?

- CA:
 - Elayne Jia
- Office hours
 - On Canvas
 - Or here:
<https://tinyurl.com/gcp-f25-oh>
- We actually have a *lot* of CAs
 - but Elayne is ours 😊

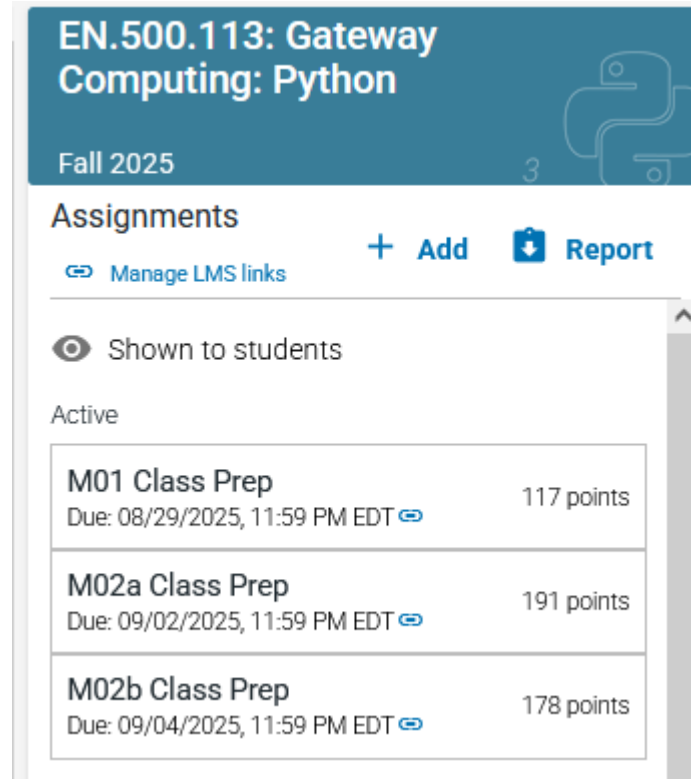


Course Overview: Where?

- IRL:
 - Krieger 309
- Online:
 - <https://jhu.instructure.com/courses/105759> (most course materials)
 - <https://learn.zybooks.com/zybook/JHUEN.500.113Fall2025/> (course textbook & class prep activities)
 - <https://www.gradescope.com/courses/1091847> (project submissions & quiz feedback)

Course Overview: When?

- Monday, Wednesday, Friday
- 10:00 -> 10:50 AM (Section 3)
11:00 -> 11:50 AM (Section 5)
- You'll have class prep due ahead of *most* classes



EN.500.113: Gateway Computing: Python

Fall 2025

Assignments

[Manage LMS links](#) [+ Add](#) [Report](#)

☒ Shown to students

Active

M01 Class Prep	117 points
Due: 08/29/2025, 11:59 PM EDT Link	
M02a Class Prep	191 points
Due: 09/02/2025, 11:59 PM EDT Link	
M02b Class Prep	178 points
Due: 09/04/2025, 11:59 PM EDT Link	

Course Overview: What?

- An Introduction to Computer Science
- You'll learn how to logically think about & reason through problems, breaking them down into smaller pieces
- You'll also learn how to use the programming language *Python*



Course Overview: What?

- Your Grade:
 - 5% - Attendance **and participation**
 - 10% - Class Prep (every day)
 - 20% - Projects (4; last one is optional)
 - 40% - Quizzes (approximately every other Friday)
 - Quizzes focus on new material, but you can't forget the basics either
 - 25% - Final Exam (date TBA)

Course Overview: Why?

- Why learn Computer Science?
 - Maybe you like the problem-solving that CS teaches
 - Maybe you want to use some computing skills in service to another major (statistics, physics, sociology)
 - Maybe you want the ability to automate things around your dorm/apartment
 - Maybe you want to create a scam cryptocurrency?
 - Please don't
 - Other reasons? You tell me

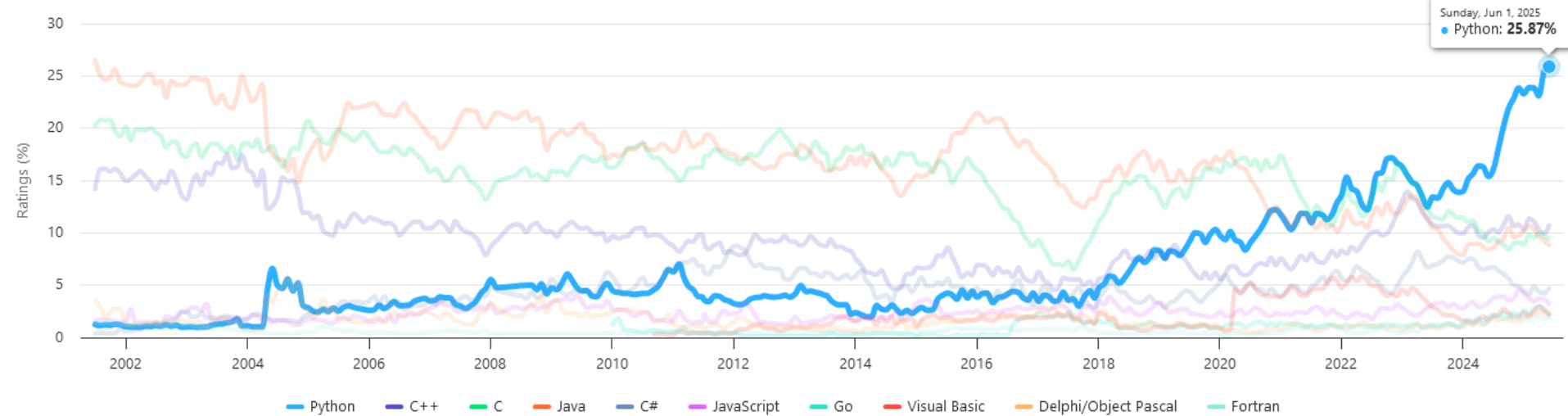
Course Overview: Why?

- Why are we learning Python in particular?
 - Python is an easy-to-use and highly expressive language
 - Easy to use, in that Python has fewer “gotchas” than languages like C
 - Highly expressive, in that you can write a little bit of code that accomplishes a lot
 - This is helped by some very powerful *libraries* – pre-written code for a certain purpose – some of which we’ll see later

Course Overview: Why?

TIOBE Programming Community Index

Source: www.tiobe.com



Course Overview: How?

- This class makes *no assumption* of any prior programming experience
 - We assume you're pretty clever, interested in learning, and know the basics of how to use a computer
 - But you are *not* expected to be familiar with Python, or any other (programming) language
 - I do expect you to know English ;)
- That being said...we will move quickly
 - Make sure to stay on top of things, and if you feel confused, *please ask*

Course Overview: How?

- Textbook: On Zybooks:
<https://learn.zybooks.com/zybook/JHUEN.500.113Fall2025/>
- We'll be using the latest version of Python 3 (3.13 at the time of this writing)
 - Realistically, for what we're doing 3.6, 3.10, and 3.12 are all equivalent
 - Just don't use Python 2 please
- If you've not installed Anaconda yet, we'll run through that in a moment

Course Overview: How?

- This is a *flipped class*
- Do not show up to class without doing the prep on Zybooks
 - Well, if you forget to do the prep, it's still better to come to class than not
 - But, you really should be doing the prep
- We will assume that you've done the prep already, and have practice activities that won't make much sense if you haven't
 - I will review the material that you've seen on Zybooks, and show a few more examples
 - But this is not intended to be comprehensive

Course Overview: How?

- Class will be dedicated to *active learning*
- I'll write code & talk you through it
- You'll reason through problems & write code
- I'll be here to answer questions & offer guidance
- Then, we'll talk through some potential solutions

Course Overview: How?

- Anaconda & Spyder
 - Anaconda is a *distribution* of Python with many useful packages built-in
 - We'll use a few of these later in the semester
 - Also built in is Spyder, a Python IDE (**I**ntegrated **D**evelopment **E**nvironment)
 - Things like a code editor, with *syntax highlighting*
 - And a Python shell

What Does Success Look Like?

- Do all of the prep ahead of class
 - This includes *starting early* too – don't wait until the morning that things are due
 - This also applies to projects
- Show up to class, and participate actively in the activities – don't just sit here
 - These in-class activities are worth a few points to encourage you to complete them
- Keep at it when things get hard – learning is hard, but this process of struggling & overcoming your difficulties is how you learn
- Solve problems yourself
 - Using the book, reference materials provided, etc is fine – but don't use Chegg, ChatGPT, etc
- Think through problems logically instead of just hacking at it
 - What I mean is, don't just try stuff at random & hope it works
 - This won't work on the exam, since you can't run any code

WHAT IS COMPUTER SCIENCE?

Activity: Find a missing card

- Your goal is to come up with a way to find which card is missing from a deck of playing cards
- Logistics:
 - Split into groups of 3
 - Each group gets a deck of cards



Activity: Find a missing card

- Within each team:
 - One person comes up with instructions
 - One person writes down instructions
 - One person manipulates the cards
- We'll go through some of your approaches in a few minutes!



Activity: Set Up Your Computer

- Download and install Anaconda from <https://anaconda.com/download>

Provide email to download Distribution

Email Address:

☐


Agree to receive communication from Anaconda regarding relevant content, products, and services. I understand that I can revoke this consent [here](#) at any time.

By continuing, I agree to Anaconda's [Privacy Policy](#) and [Terms of Service](#).

Submit >

Skip registration

Distribution Installers

 Download

For installation assistance, refer to [troubleshooting](#).



Windows



Mac



Linux



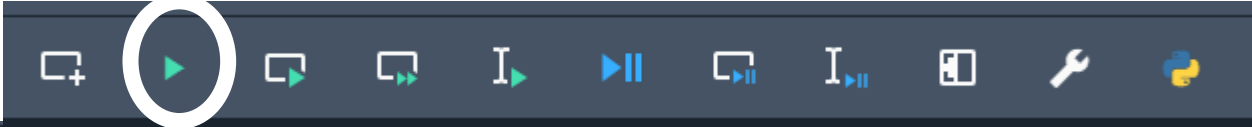
(it'll probably look slightly different on the Mac, but similar idea)

Activity: Test Spyder

- Make sure Spyder works

1

2



The screenshot shows the Spyder IDE interface. The top toolbar contains several icons for file operations and execution. The 'Run' icon, which is a green play button, is circled with a white circle and labeled with the number '2'. To the left of the toolbar, the number '1' is displayed. Below the toolbar, the file explorer on the left shows a file named 'temp.py' with a tab icon and a close button. The main editor window displays the contents of 'temp.py', which is a Python script with a docstring and a print statement. The console on the right shows the output of the script, which is 'Hello from Python!'. The number '3' is displayed at the bottom right of the console area.

```
C:\Users\Kai\.spyder-py3\temp.py
```

```
1 # -*- coding: utf-8 -*-
2 """
3 Spyder Editor
4
5 This is a temporary script file.
6 """
7
8 print("Hello from Python!")
```

```
A X
```

```
| packaged by Anaconda, Inc. | (main, Jun 12 2025, 16:
t", "credits" or "license" for more information.

-- An enhanced Interactive Python. Type '?' for help.
```

```
In [1]: %runfile C:/Users/Kai/.spyder-py3/temp.py --wdir
Hello from Python!

In [2]:
```

3

Activity: Enroll in ZyBooks

- Class prep is through Zybooks
- Please get enrolled in the course there:
<https://learn.zybooks.com/zybook/JHUEN.500.113Fall2025/>