

■ Self-introductions

- I'm Jaime Abbariao
- Graduated from Baruch in 2017 and NYU in 2020
- Currently a software engineer at Figma (<https://www.figma.com>)

■ About this class

- We meet on Tuesday and Thursday at 5:40pm in 6-125
- Class will be fast-paced
- Mixed focus on implementing math-related problems and practical tools

Syllabus

Course grading

- Problem sets: 50%
- Midterm: 25%
- Final: 25%

If your final exam grade is higher than your midterm exam grade, then I'll substitute your final exam grade for your midterm grade.

Exams will generally be cumulative in nature because we're going to continue building your base knowledge throughout the semester.

Grade breakdown

- A: 95-100
- A-: 90-94
- B+: 87-89
- B: 83-86
- B-: 80-82
- C+: 77-79
- C: 73-76
- C-: 70-72
- D: 65-69
- F: 0-64

These grades are subject to change (in your favor). This is my first time teaching a class so depending on how the class does on exams, I'll communicate how these grades will change.

■ Syllabus

■ Problem sets

I just want to state some policies around our problem sets.

- Late submissions are penalized 5 points every 3 hours past the deadline.
 - Ex. If the deadline is 12am, but you submit at 6am, your max possible score is a 90 for the homework.
- All non-programming questions in problem sets should be typed up and submitted as a PDF.
- Feel free to collaborate on homework with your classmates, but you must note who in your homework submission.
 - We'll limit this to groups of 2!

■ A note about the use of AI tools

It's inevitable that you may end up using AI to help you with your problem sets.

I really can't do much to stop you. Given that I use AI in my day job, it feels a bit hypocritical to tell you to stop.


To combat this, the midterm and final will be paper-based exams.

■ Textbooks

There is no official textbook.

If you need a reference guide, then you should look into Python Crash Course by Eric Matthes.

 Syllabus

 Contact

If you need to contact me about any questions or concerns, you can reach me at jaimeabbariao@gmail.com

Just make sure that you prepend your email subject with **MTH 3300**.

■ Tips for doing well in this class

- Practice everyday
 - You will not do well on the exams if you do not practice coding
- Ask questions early and often
- Collaborate with your classmates

■ What is programming?

Programming is the process of creating instructions that a computer can follow to perform specific tasks.

A computer program follows an algorithm which is a set of instructions intended to solve some problem.

■ Pseudocode

We can express algorithms in pseudocode which is notation that resembles a programming language.

For example, let's say that we want to write a program that prints "Fizz" if a number is divisible by 3, "Buzz" if a number is divisible by 5, and "FizzBuzz" if the number is divisible by both 3 and 5.

How can we solve this with pseudocode?

```
if number is divisible by 3 and number is divisible by 4 then
  print "FizzBuzz"
else if number is divisible by 3 then
  print "Fizz"
else if number is divisible by 5 then
  print "Buzz"
end
```

The important aspect about pseudocode is that you're expressing your thoughts in a manner that is structurally similar to how you're going to write code.

This is going to be an important aspect during exams! If you forget the syntax for certain things, but you can show that you know how to express your ideas with pseudocode, you will earn points.

■ What is programming?

■ Programming languages

Obviously, we cannot run computer programs with just pseudocode.

In order to create computer programs that can be transformed into machine language (binary), we need a human-readable language.

Python was created by Guido van Rossum and released to the public in 1991.

Python has positioned itself as a great introductory programming language for the following reasons:

- Ease of learning: Python has a simple and readable syntax which is designed to be easy to understand and write.
- Relevance: Python is widely used in the industry.
- Enforces good habits: Python encourages good programming practices like readability and simplicity

■ Introduction to Python

■ Hello, World!

Here's our canonical introduction to Python!

```
print("Hello, world!")
```

■ Installing Python

This class requires that we have at least python@3.10

■ For MacOS users

If you don't already have Homebrew, please following the instructions on the site (<https://docs.brew.sh/Installation>) to install it.

```
brew install python@3.12
```

Run the following command to make sure that you've successfully installed the correct version

```
python --version
```

■ For Windows users

Visit the official download page (<https://www.python.org/downloads/>) and download Python 3.12

Run the installer and make sure to add Python to your path.

Run the following command to make sure that you've successfully installed the correct version

```
python --version
```

■ For Linux users

If you don't have at least Python 3.10 installed on your machine, you can do one of two things:

- Use your distribution's package manager to download the appropriate version
- You can download the source code from the Python site and build it. Handy link here if you need it (<https://docs.python.org/3/using/unix.html>)

■ Using the Python interpreter

Before we create our first files, the easiest way to run Python is by running the following command in your terminal.

```
python
```

Calling `python` opens up the interpreter which allows us to have an immediate feedback loop.

To exit the interpreter, you can type in `exit()` or use `ctrl + d`

Exercise: Use the interpreter to print out your name

■ Code editors

Here are a list of code editors that I would recommend for beginners:

- Visual Studio Code (<https://code.visualstudio.com/>) (recommended)
- PyCharm (<https://www.jetbrains.com/pycharm/>)
- Zed (<https://zed.dev/>)
- neovim (<https://neovim.io/>)

For those that want a simple setup, VSCode will be enough.

NOTE: I do prefer students to use code editors over Jupyter notebooks because you will need to upload actual Python files for your assignments

■ Running your first Python file

■ add.py

```
def add(a, b):  
    return a + b  
  
if __name__ == "__main__":  
    a = int(input("Enter one number: "))  
    b = int(input("Enter another number: "))  
  
    print(add(a, b))
```

If you're using Visual Studio Code or PyCharm, there should be an option to run your code if you have the appropriate plugins.

If want to run this code in the terminal, run your file with the following command:

```
python <path-to-file>/add.py
```