MTH 3300 STRA Spring 2025

Tu,Thu 5:40pm - 7:20pm, B 6-125

Instructor: Jaime Abbariao, Software Engineer @ Figma

Contact: Feel free to send questions to *jaimeabbariao@gmail.com*.

Office hours: Fridays, 5:30pm - 6:30pm on Zoom

Course overview

This course is designed to provider undergraduate students with a solid foundation in programming using Python.

Students will learn fundamental programming concepts, problem-solving skills, and practical applications of Python in various domains.

Course outline

Module 1: Introduction to Programming

- What is programming?
- Overview of Python and its applications.
- Setting up the environment (IDE, installation, etc.).
- Writing and running your first Python program.
- Variables, data types, and basic input/output.
- Operators (arithmetic, relational, logical).

Module 2: Control Structures

- Decision-making: if, elif, else statements.
- Boolean expressions.
- Loops: for and while loops.
- Loop control statements (break, continue, pass).

Module 3: Data Structures

- Introduction to lists, tuples, and sets.
- List operations and slicing.
- Simple applications: creating and manipulating collections of data.
- Dictionaries: keys, values, and operations.

Module 4: Functions and Modular Programming

- Defining and calling functions.
- Parameters, arguments, and return values.
- Scope and lifetime of variables.
- Writing modular programs using functions.
- Introduction to recursion.

Module 5: Working with Files and Error Handling

- Reading from and writing to files.
- Handling different file formats (e.g., text and CSV).
- Practical exercises: creating logs, processing data files.
- Introduction to error handling with try, except, finally.
- Debugging techniques and tools.
- Writing robust programs with error handling.

Module 6: Object-oriented Programming

- Introduction to OOP concepts: classes, objects, attributes, and methods.
- Creating and using classes and objects in Python.
- Advanced OOP: inheritance, polymorphism, encapsulation, and abstraction.

Assessment and grading

- Problem sets: 40%
- Final project: 10%
- Midterm: 25%
- Final Exam: 25%

Homework policy

- 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!

Tips on doing well in class

- Practice. Practice. Practice. You will not learn how to code by just looking through my slides.
- There are no stupid questions. If you're lost, please ask. It's highly likely that someone has the same question you do.
- Collaborate! It's okay to ask each other for help (except on exams).