Syllabus (Fall 2023)

Course Information

Catalog Number	56:219:500 (also cross-listed as 56:198:500 and 56:121:531)
Instructor	Sunil Shende
Schedule	**MW from 3:45pm to 5:05pm**
Classroom	**CS-109** Cooper Classroom Building
Office Hours	On Zoom **MW from 10am to 11am**
Email	shende AT camden DOT rutgers DOT edu

Learning Outcomes

Students are not expected to have prior programming experience. In general, by the end of the course, it is expected that successful students will:

- understand and use appropriate programming constructs for developing Python code
- document and test code cleanly and robustly
- apply computational and inferential thinking in program construction
- use specific built-in Python libraries and third-party libraries for data science for computational problem solving

Textbook

We will use the following book as our primary textbook:

> Introduction to Computation and Programming Using Python, third Edition: With Application to Computational Modeling and Understanding Data, by John V. Guttag. Published by The MIT Press, 2021.

You can buy the book but it is also available **online** from the Rutgers Libraries site: please use your Rutgers NetID to login and access it. Please **do not download and print** book chapters - save the planet by not wasting unnecessary paper!

As a way of supplementing and complementing material from the textbook, I will also provide additional notes, links to topics on the internet or links to relevant videos.

Logistics

We will be using the Rutgers **Canvas** site for the course as well as Github or Google Drive to share class material. The class will run in **semi-flipped** mode as follows:

- on Wednesdays, we will have a traditional lecture alongside programming and problem solving, and
- on Mondays, you are expected to arrive to class having already read assigned readings and watched assigned videos (if any) for that week. In class, you will work on specific programming programming problems as a means to develop mastery, and you will submit these programs for credit.

Tentative Schedule of Topics

Please note that this schedule is **tentative** and subject to change. The weeks refer to two consecutive class sessions and not to calendar weeks, e.g. Week 1 below includes Wednesday (Sep 6) and Monday (Sep 11).

Week	**Topics**
1	Introduction: Variables, Datatypes, Objects; Environments
2	Flow of Control: Loops, Conditionals and Matching
3	Functions; Exceptions; Documenting code
4	Functional abstraction; Namespaces; Lists and Tuples
5	Sequences; Dictionaries; Iterables
6	Reading and writing files; Testing code
7	Object-oriented programming: Classes
8	Using built-in classes and designing classes
9	Special methods; Iterators & Generators; Protocols
10	Data Science with `numpy`, `scipy`, `pandas`
11	Plotting and Visualizing Data: `matplotlib`
12	Randomness, Sampling and Distributions
13	Simulation, Hypothesis Testing
14	Review

Grading

The overall grade will be apportioned based on the following assessments:

- **30%** Bi-weekly homework (drop worst grade out of 7)
- **30%** Weekly In-Class Assessments (omit worst 2 grades)
- **40%** Two in-class midterm exams

Except in the most extenuating circumstances, there will be **no makeup** opportunities for exams or in-class assessments. Late submission of homework problem sets will generally **not be entertained** except in the case of a verifiable, documented emergency (medical or personal). If you require special accommodation, e.g., extra time on the exams or in-class assessments, please obtain appropriate documents from the Division of Student Affairs.

Letter grade rubric

I generally use (with some minor variation at my discretion) the following rubric:

- an A grade for overall points above 85%
- a B+ grade in the range 75 – 85%
- a B grade in the range 65 – 75%
- a C+ grade in the range 60 – 65%
- a C grade in the range 50 – 60%

Anything below 50% is considered an unsatisfactory or failing grade (either a D or F grade).

Academic Integrity

Some of the in-class assessments will be collaborative, while others will need to be completed individually. There are lots of places and mechanisms whereby you can get help to solve your homework problems, but be warned that there are many gaps in reasoning and incorrect solutions in the wild (including code that you may find on websites or after consulting your favorite large language model). If challenged, you will have to be prepared to clearly explain your code and/or written submissions to my satisfaction. This is a graduate course, and a large part of being successful in graduate school lies in recognizing that learning does not come from mimicry or plagiarism.

I strongly encourage you to learn from other books, internet resources like StackOverflow, and even by prompting LLMs like GPT-X/ CoPilot/Bard. But, you must **cite your sources** at all times! Copying from someone or somewhere without citation or *simply allowing* your work to be copied by others constitutes cheating, as does *blind transcription* from sources including books, LLM transcripts and the internet at large. *You* are ultimately responsible for what you turn in: if it is determined that your work (even with citations) is derivative and you haven't completely understood what you have submitted, it will count as an academic integrity violation and will carry pretty serious penalties.

I will follow the Rutgers Academic Integrity Policy and Student Code of Conduct to deal with suspected violations. Please read and understand the policy carefully.