



## Data Focused Python 95-888

**Location:** HBH 1204

**Semester:** Fall Mini, **Year:** 2024

**Units:** 6, **Section(s):** B1, **Class Times:** Tues and Thurs 17:00-18:20

---

### Instructor Information

**Name** Michael Simko – “Mike”

**Contact Info** [msimko2@andrew.cmu.edu](mailto:msimko2@andrew.cmu.edu)

### TA Information

**TA name** TBD

**TA Contact Info**

**Office location**

**Office hours**

---

### Course Description

- This course focuses on the fundamentals of computer programming using the Python 3 language. Students will develop their problem-solving skills using a top-down approach to build real-world based software applications and will also learn the basics of the software lifecycle: planning, development, testing, implementation and maintenance. Assignments will include homework, two checkpoint quizzes and a capstone data focused programming project. Learners will study how to build professional, user-friendly computer programs to real-world applications in an IT environment.

### Learning Objectives

- Use a Python IDE (integrated development environment) in interactive and script mode to both test code snippets and author professional programs.
- Learn command line usage for executing Python source code.
- Develop problem-solving skills through practice and understanding of the top-down approach.
- Form and manipulate collections of data: list, tuple, set, dict, NumPy ndarray, and Pandas Series and DataFrame.
- Produce modules of function definitions for code reuse.
- Create object-based algorithms to solve real-world problems using the Python language.
- Be exposed to the SDLC (software development lifecycle) to understand how software applications are authored in industry.

## Learning Resources

No textbooks are required for the course, but some good resources include:

- Wes McKinney, Python for Data Analysis (<https://wesmckinney.com/book/>)
- Al Sweigart, Automate the Boring Stuff (<https://automatetheboringstuff.com/>)

Additionally, students will be presented with slides, online tutorials, and other materials. Please keep an eye on our CANVAS page for these resources.

There are many resources online to help when learning Python. A few that are particularly relevant for this course are listed below.

- VSCode (<https://code.visualstudio.com/>)
- Jupyter Notebooks Documentation (<https://jupyter.org/>)
- Numpy documentation (<https://numpy.org/>)
- Pandas documentation (<https://pandas.pydata.org/pandas-docs/stable/index.html>)
- Matplotlib documentation (<https://matplotlib.org/>)

## Course Work

Your grade in this course will be determined by a series of 5 weekly homework assignments (25%), in class quizzes (20%) and a final project (55%).

## Assignments

Weekly assignments will take the form of a Python file: namely, code snippets integrated with captions and other narrative. All assignments are due **on the date posted on Canvas, usually before midnight (Pittsburgh/eastern US time) before the next week's class session**. Each homework assignment may have several parts. Your score for each assignment will be calculated according to the scheme outlined in the rubric below.

## Homework Rubric

### Total: 5 points

**Correctness:** Each homework assignment will be worth 10 points total. Deductions will be made at the discretion of the grader.

**Style:** Coding style is very important. With the exception of Homework 1, you will receive a deduction of up to **1 point** if you do not adhere to good coding style.

- No deduction if your homework is submitted with:
  - good, consistent coding style
  - appropriate use of variables
  - appropriate use of functions
  - good code commenting
  - correct choice of variable names
- - **0.5** if coding style is acceptable, but fails on a couple of the criteria above.
- - **1** if coding style is overall poor and fails to adhere to many of the above criteria.

## Quizzes

There will be 2 short in-class quizzes scheduled during the later weeks of class worth 10 points each. Dates and times will be announced in advance. The purpose of these quizzes is to assess understanding of various concepts that are central to the class. You will have access to the course materials, online resources and whatever resources you need to answer the questions. However, you ***may NOT collaborate*** with others and your answers must be your own. Your score on these quizzes will count for 20% of your final grade.

## Final project

The final Project consists of designing, developing, and presenting a Python-based application of interest to some customer base. You can think of your product as a commercial product, or as a non-profit public service product. Your application will need to scrape or otherwise obtain data from an online source, but may also include some element of static datasets. Your product will then need to clean, organize, and analyze the obtained data to produce interesting results for the user to look at (statistical results, tables, plots, ...) and perhaps interact with. Since this is a Python course, all of this will need to be in Python code: you may not download data using a browser, clean and organize the data in Excel, and then just do the analysis and display part in Python. Your entire product must be Python from beginning to end.

A separate rubric exists to more detail the final project requirements which will be on Canvas, and will be discussed extensively during class time.

**Regardless of grading basis, students must receive a score of at least 50% on the final project in order to pass the class.**

Grade	Percentage Interval
A+	100-97% Outstanding
A	96-93% Excellent
A-	92-90% Very Good
B+	89-87% Good
B	86-83% Acceptable
B-	82-80% Fair
C+	79-77% Poor
C	76-73% Very Poor
C-	72-70% Minimal Passing
Fail	<70%

## Course Grading

Your final course grade will be calculated according to the following breakdown.

HW Assignments	25%
Quizzes	20%
Final project	55%

Late submissions

Homework is to be submitted on the due date indicated (posted on Canvas as mentioned above). **Late homework will NOT be accepted for credit.**

## Collaboration

You are encouraged to discuss homework, quizzes and project work, with your fellow students and as a group during class sessions. However, **any work you submit must be your own**. You must acknowledge in your submission any help received on your assignments. **That is, you must include a comment in your homework submission that clearly states the name of the resource or reference from which you received assistance.** Submissions that fail to properly acknowledge help from other students or non-class sources **will receive no credit**. Copied work **will receive no credit**. Any and all violations **will be reported** to Heinz College administration.

All students are expected to comply with the CMU policy on academic integrity. This policy can be found online at <http://www.cmu.edu/academic-integrity/>.

What constitutes plagiarism in a coding class?

The course collaboration policy allows you to discuss the problems with other students, but requires that you complete the work on your own. Every line of text and line of code that you submit must be written by you personally. You may not refer to another student's code, or a "common set of code" while writing your own code. You may, of course, copy/modify lines of code that you saw in any lecture or during in-class "lab work".

The following discussion of code copying is taken from the [Computer Science and Engineering Department at the University of Washington](#). You may find this discussion helpful in understanding the bounds of the collaboration policy.

"[It is] important to make sure that the assistance you receive consists of general advice that does not cross the boundary into using code or answers written by someone else. It is fine to discuss ideas and strategies, but you should be careful to write your programs on your own."

"You must not share actual program code with other students. In particular, you should not ask anyone to give you a copy of their code or, conversely, give your code to another student who asks you for it; nor should you post your solutions on the web, in public repositories, or any other publicly accessible place. [You may not work out a full communal solution on a whiteboard/blackboard/paper and then transcribe the communal code for your submission.] Similarly, you should not discuss your algorithmic strategies to such an extent that you and your collaborators end up turning in [essentially] the same code. Discuss ideas together, but do the coding on your own."

"Modifying code or other artifacts does not make it your own. In many cases, students take deliberate measures -- rewriting comments, changing variable names, and so forth -- to disguise the fact that their work is copied from someone else. It is still not your work. Despite such cosmetic changes, similarities between student solutions are easy to detect. Programming style is highly idiosyncratic, and the chance that two submissions would be the same except for changes of the sort made easy by a text editor is vanishingly small. In addition to solutions from previous

years or from other students, you may come across helpful code on the Internet or from other sources outside the class. Modifying it does not make it yours."

"[I] allow exceptions in certain obvious instances. For example, you might be assigned to work with a project team. In that case, developing a solution as a team is expected. The instructor might also give you starter code, or permit use of local libraries. Anything which the instructor explicitly gives you doesn't normally need to be cited. Likewise, help you receive from course staff doesn't need to be cited."

## Generative AI

Like any other online resource, generative AI (ChatGPT, Google Bard, Bing, etc) are allowed as additional tools in this class for learning and generating correct Python code. However, your responsibilities as a student remain the same – **you must follow the academic integrity guidelines of the university and the guidance for citations given above**. If you use any generative AI, you are required to cite the tool's contributions to your final work – using these resources without proper citation is considered plagiarism. Ultimately, **you are responsible for the content that you submit!**

If you have any questions about any of the course policies, please don't hesitate to ask directly.

## Course Policies

- **Attendance & Participation:** Regular class attendance is expected, although accommodations can be made for absences with good reason. Questions during class and discussions are **strongly encouraged**. The best way to learn Python is by creating code, making mistakes, and finding ways to produce results. Course time will be interactive with students encouraged to use their laptops/computers to connect and participate in the course as well as run Python code and follow along with real time exercises. Course information in the form of slides will usually be shared before the start of class, and any questions about the material can be asked during the class session. There will be exercises during some of the classes and you will be asked to code along and participate by answering questions and offering code syntax and format.
- **Accommodations for students with disabilities:** If you have a disability and require accommodations, please contact Catherine Getchell, Director of Disability Resources, 412-268-6121, [getchell@cmu.edu](mailto:getchell@cmu.edu). If you have an accommodations letter from the Disability Resources office, I encourage you to discuss your accommodations and needs as early in the semester as possible. We will work with you to ensure that accommodations are provided as appropriate.
- **Statement on student wellness:** As a student, you may experience a range of challenges that can interfere with learning, such as strained relationships, increased anxiety, substance use, feeling down, difficulty concentrating and/or lack of motivation. These mental health concerns or stressful events may diminish your academic performance and/or reduce your ability to participate in daily activities. CMU services are available, and treatment does work. You can learn more about confidential mental health services available on campus at: <http://www.cmu.edu/counseling/>. Support is always available (24/7) from Counseling and Psychological Services: 412-268-2922.
- **Mobile Devices:** Mobile devices are discouraged during class time. When live, and in person on campus, the use of phones in class is prohibited. If you need to take a call, you may be excused and leave the room, but avoid this whenever possible. Focus should be on the class during the in-person sessions.

## Course Schedule

Week	Theme/Topic	Learning Outcomes Addressed	Assignments Due
1	Introduction to Python, IDEs and basic data types		none
2	More on data types, variables, structures and operations, packages and basic file handling		HW#1 - first Python script
3	Web scraping		HW#2
4	Control structures, functions and modules		HW#3 and first quiz
5	Graphics		HW#4
6	Object Oriented Programming (OOP)		HW#5 and final quiz
7	Advanced topics and final project discussions		none