Introductory Computer Programming for METOC (MR2020; Summer Quarter 2022)
Officially Computer Computations for Air-Ocean Sciences
Instructor: Scott Powell (Root 255)
Room: IDEA Lab (Root 123)
Meeting times: M, W: 1500–1650
Course webpage: https://swpowell.github.io/MR2020.html

Course Objectives

- Learn to use and navigate UNIX based systems (e.g., Mac, Linux).
- Learn fundamentals of using computer logic to translate an outline of desired operations into computer code
- Familiarization with developing code in the python programming language to accomplish common tasks such as and reading/writing data, executing computations on that data, and plotting one-dimensional and multi-dimensional data
- Build your own code from scratch to read in data and generate your own plots in the form of a final project.

Syllabus

The syllabus is approximate. We will plan to cover all material listed, but the timing may differ from the schedule depending on the learning pace of the class. In addition, there may be material covered that is not listed on the syllabus depending on the needs of students and on questions raised during class. For Weeks 1–4, YouTube material should be viewed before class.

Week 1 (July 6): Getting setup on IDEA lab or personal computers.
Homework: Review Visual Studio Code Introduction (https://code.visualstudio.com/docs/introvideos/basics)

Week 2 (July 11, 13): Running a simple code. Data Types (numbers, strings, lists, tuples, sets, dictionaries)
NOTE: Go through code during class time on Monday. Instructor will be on leave.
YouTube: Python Crash Course (19:30 through 1:19:24)

Week 3 (July 18, 20): Control structures: If-else blocks. For and while loops. Try-except blocks. Match-case blocks.
YouTube: Python Crash Course (1:19:29 through 1:55:09)
Homework: Problem Set 1 (Due July 25)

Week 4 (July 25, 27): Modules. Functions. Classes. Errors and Exceptions.
YouTube: Python Crash Course (1:55:14 through 2:27:21)

Week 5 (Aug. 1, 3): Numpy arrays and functions. Indexing. File input/output

Homework: Problem Set 2 (Due Aug. 8)

Week 6 (Aug. 8, 10): Midterm week

NOTE: Instructor on official travel Monday and Wednesday. Monday would be a good day to work through practice midterm during class time.

Midterm Exam: Aug. 10 at 1500

Week 7 (Aug. 15, 17): Introduction to pandas. Parallel coding using multiprocessing module.

Weeks 8–9 (Aug. 22, 24, 29, 31): Create figures and subplots, making line plots, scatter plots, histograms, two-dimensional plots. Plotting geospatially located data on maps. Labeling axes. Creating legends and colorbars.

Weeks 10–11 (Sept. 6, 7, 13, 14): Final coding practicum. Instructor will be in classroom during regular meeting hours plus additional time for one day during finals period.

NOTE: Tuesday is a Monday schedule for Labor Day.

Grading

Midterm (50%)
Final practicum (50%)

All students will have two opportunities to take the midterm exam. Any student who is not satisfied with their grade on the first try will be allowed to attempt a second different exam covering the same topics. The highest of the two grades will be the one recorded.

Approximate grading scale (after highest grade is curved to 100*):

These are the minimum letter grades students can receive if their numerical grade is within the range shown in the left column. For example, a student with a final grade of 90 will have earned no lower than a A-.

Numerical Grade	**Minimum Possible Letter Grade**
93–100	A
90–92	A-
87–89	B+
83–86	B
80–82	B-
77–79	C+
73–76	C

*The difference between 100 and the highest student grade at the end of the course will be added to each student's grade before being converted to a letter grade. For example, if the highest course grade is 93, 7 points will be added to each student's grade. The new total with the additional 7 points will then be converted to a letter grade.

70–72	C-
67–69	D+
63–66	D
60–62	D-
Below 60	X/F

Course Structure:

1. Course material will be available at or linked from the course webpage.

2. The Teams chat group for MR2020 is the venue for group discussion and questions. Students may also reach the instructor via private chat on Teams or email.

3. The final project will be graded based on the following:

- Did the student generate the correct final product/output and can the code be run by the instructor? (50%)
- Is the code readable, properly structured (i.e., proper use of functions, etc.) and well-commented? (30%)
- Is the code efficient (i.e., run time and memory usage minimized)? (20%)