SD 212: Data Science & Programming II

Course Policy, Spring AY2026

Instructors:

- Prof. Daniel S. Roche, 457 Hopper Hall, x36814, roche@usna.edu (Coordinator)
- LT Brett Gentile, 458 Hopper Hall, x36572, bgentile@usna.edu

MGSP Leaders:

- Maggie Pigott (32nd co), m275112@usna.edu
- Luke Roskelley (11th co), m275490@usna.edu

Course Description: This course builds on the programming skills developed in the prerequisite course and moves the focus towards a wider software ecosystem in order to solve more complex data science tasks. Students will learn and apply foundational principles of program organization including classes and objects, interfaces, inheritance, abstraction, and decoupling. In addition, important command-line skills will be developed for data gathering and cleaning, as well as library and software acquisition and use. These principles will be utilized through high-level programming in Python to analyze and manipulate real-world data sets.

Credits: 3-2-4

Prerequisites: SD211 Intro to Data Science and Programming

Learning Objectives:

1. Observe the structure of new datasets and perform basic data cleaning and manipulation using command-line tools. (supports outcome 2)
2. Understand how regular expressions can be used to describe tokens and their use in programs to manipulate plain-text inputs. (supports outcome 1)
3. Write programs to analyze real datasets using popular data science libraries. (supports outcome DS-6)
4. Utilize basic object-oriented principles such as inheritance and operator overloading to develop and structure complex programs. (supports outcome 1)
5. Understand how libraries are packaged, distributed, downloaded, and installed using standard tools.
6. Examine how data science has been and can be used to impact society at large. (supports outcome 4)

Student Outcomes: Graduates of the program will have an ability to:

1. **Analysis.** Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions.
2. **Implementation.** Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline.
3. **Communication.** Communicate effectively in a variety of professional contexts.
4. **Ethics.** Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.
5. **Teamwork.** Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline.

DS-6. Data. Apply theory, techniques, and tools throughout the data analysis lifecycle and employ the resulting knowledge to satisfy stakeholders' needs

Syllabus:

- **Unit 1: Welcome back** (Classes 1–3)
  Course overview, Data science pipeline, Python review
- **Unit 2: OOP in Python** (Classes 4–9)
  Operator overloading, Inheritance
- **Unit 3: Command line** (Classes 10–13)
  Files and directories, bash commands, Piping and redirection
- **Unit 4: Regular expressions** (Classes 14–16)
  Regex syntax, Python re, Command-line tools
- **Unit 5: Error handling** (Classes 17–20)
  try/except, return codes, bash if statement
- **Unit 6: Versions and packaging** (Classes 21–22)
  git, github, pip, mamba
- **Unit 7: Data cleaning** (Classes 23–24)
  Missing data, Outliers, Preprocessing, Merging dataframes
- **Unit 8: Hardware and OS** (Classes 25–27)
  CPU, Memory hierarchy, Filesystems, Compiled vs Interpreted code, Role of the operating system
- **Unit 9: Concurrency** (Classes 28–32)
  Multithreading, Python GIL, Multiprocessing, bash job control
- **Unit 10: Machine learning with sklearn** (Classes 33–36)
  Statistical data types, Reading documentation, Classification, Regression, Clustering
- **Unit 11: Data Ethics** (Classes 37–39)
  Principles, Case studies

Updates to the course policy: In case this course policy needs to be changed during the semester, students will be notified by email and verbally during class. The current version will always be posted on the course website.

Textbooks:

- Charles Severance. *Python for Everybody: Exploring Data in Python 3*, online, 2023.

- Wes McKinney. *Python for Data Analysis*. O'Reilly Media, 3rd ed., 2022.
- William E. Shotts. *The Linux Command Line* No Starch Press, 2nd ed., 2019.
- Jeroen Janssens. *Data Science at the Command Line*, O'Reilly Media, 2nd ed., 2021.
- Mendel Cooper. *Advanced Bash-Scripting Guide*, Public domain, rev. 10, 2014.
- Alex Martelli, Anna Martelli Ravenscroft, Steve Holden, Paul McGuire. *Python in a Nutshell*, O'Reilly Media, 4th ed., 2023.
- Jake VanderPlas. *Python Data Science Handbook*, O'Reilly Media, 2nd ed., 2022.
- Suzanne J. Matthews, Tia Newhall, Kevin C. Webb. *Dive Into Systems*, No Starch Press, 2022.

Course Website: https://roche.work/courses/s26sd212/

Extra Instruction: Extra instruction (EI) is strongly encouraged and is most easily scheduled by email. EI is not a substitute lecture; students should come prepared with specific questions or problems.

Collaboration:

The guidance in the Honor Concept of the Brigade of Midshipmen and the Computer Science Department Honor Policy must be followed at all times. See https://www.usna.edu/CS/resources/honor.php. Specific instructions for this course:

- Collaboration or assistance from any human other than the instructors, MGSP leaders, and those enrolled in SD212 this semester is not permitted. To be crystal clear, this prohibits assistance from prior enrolled midshipmen. This includes any written or electronic materials from previous semesters.
- Homework: Students may collaborate on homework with others in the same class, but must cite this collaboration clearly. Every student must actually complete their own assignment and understand anything they turn in.
- Labs: Each lab presents a significant challenge and opportunity to develop and demonstrate mastery. The goal is not to simply complete the assignment and get the right answer, but to actually engage in the process to develop that answer through the data science skills we are learning in SD212.

  Unless otherwise specified, for labs:

  - Discussion of general strategies, tools, and tips is allowed (and encouraged) between current SD212 students. Examples: "How do I get pandas to read in dates correctly?" or "What web page did you use to figure out how to make that graph?"
  - Sharing specific solutions (such as source code) is **not** permitted. Example: "How did you do part 3?" or "Here is the for loop I used to convert the data".

  - Looking at a fellow student's code to help them debug it is allowed *after you have already solved that part yourself.* Example: Lucy has finished part 1 of the lab. Steve has tried to write a solution for part 1 but it's giving an error or not working. Lucy looks at Steve's code and offers some suggestions on how to fix it.
  - Looking at someone else's code when you have not yet completed that part is **not** allowed. Example: Lucy has finished the lab and Steve is still working on part 1. Lucy lets Steve look at her code for part 2 to see how she solved it.
  - When in doubt, **ask your instructor**. We are all on the same team and trying to become better data scientists. Your instructor wants to help you succeed and is not trying to trap or trick you. We also know that you are just learning and struggling and don't expect you do do everything perfectly the first time.

- Quizzes and Exams: No collaboration is allowed. Any group study guides should be shared with the instructor. For practicum quizzes and exams, the only websites that may be used are the course website and official documentation for tools and libraries such as <python.org> that is linked from the course website.. Definitely no web searches, social media sites, or AI tools may be accessed.

All collaboration and outside sources should always be cited. The same rules apply for giving and receiving assistance. If you are unsure whether a certain kind of assistance or collaboration is permitted, you should assume it is not, work individually, and seek clarification from your instructor.

Use of Generative AI: The use of generative AI tools, such as Copilot and ChatGPT, to help complete assignments is treated the same as collaboration or assistance with a human (see above) and is therefore prohibited under most circumstances.

The **only** exception is the course-specific Gemini "Gem", which can be used for studying, homeworks, and labs, as long as any code you use is clearly documented in what you turn in.

For written aspects (not programming) of assignments, the same prohibition of AI tools is in place. All written portions must be your own. The only exception is the use of grammar checkers. Grammar checkers, such as Grammarly, are allowed for use with "fixing" your already written grammar and syntax, but you may not use its text generation capabilities. When in doubt, ask your instructor.

Classroom Conduct:

Everyone in the classroom will show appropriate respect to each other at all times.

This class relies on active engagement and frequent interaction. Use of electronic devices during class time outside of note-taking apps is not permitted.

The section leader is responsible for recording attendance, bringing the class to attention, notifying the CS department office if the instructor is more than 5 minutes late, and directing the class in useful work in the instructor's absence.

Drinks are permitted, but they must be in closable containers. Food, alcohol, and tobacco (of all kinds) are prohibited.

Electronic devices must be silent during class and should never serve as a distraction to other students.

USNA follows the practice of non-attribution to protect all faculty, staff, midshipmen, and guests. To foster an environment where ideas are openly exchanged, this class follows the practice of non-attribution for all communications (in-person, written, and electronic). If you wish to refer to another person's ideas or comments outside of this class, you may refer to them as "a fellow class member" or "a speaker," but you may not disclose the speaker's identity without their express permission.

Absences:

Students are responsible for all class material. Notes will be posted for each lecture, along with recommended readings. However, this material is not exhaustive and students missing class should arrange to copy notes from a classmate.

Late Policy:

Homework solutions will generally be discussed immediately, and so no late submissions of homeworks will be accepted for credit. The same deadline applies even in the case of excused absences; students who will miss class should ensure that their work is still submitted on time (typically, electronically).

Labs: Each student has up to 3 *grace days* they may use at their discretion at any point during the semester for lab deadlines (*either* milestone or final deadline). Each grace day extends 1 deadline for 1 student by 24 hours.

Grading:

The work of the class consists of:

- Homeworks (2-3 per week)
- Labs (every 2 weeks)
- Quizzes (during lab time every week)
- Midterm exams (6-week and 12-week)
- Final exam

Most quizzes, as well as a portion of all three exams, will include a practicum component, to be completed and submitted electronically without the use of any AI tools or websites other than those specified.

Any student that completes every homework assignment to a satisfactory level will have their two lowest homework grades dropped at the end of the semester. The definition of "satisfactory level" is based on effort and is at the sole discretion of the instructor. *Work submitted late may count for this requirement, even if it is late and gets zero credit.*

Plus/minus grades will be assigned based on the following numerical cutoffs.

	-		+
A	90–92	93–100	
B	80–82	83–86	87–89
C	70–72	73–76	77–79
D		60–66	67–69
F		0–59	

Grade percentages will be rounded down to the nearest integer. As an example, a percentage of 89.7% is a B+.

Term grades will be assigned using the following weights:

	6 weeks	12 weeks	16 weeks	Final
Homeworks	20%	15%	15%	15%
Labs	30%	30%	30%	30%
Quizzes	20%	15%	15%	10%
Midterms	30%	40%	40%	20%
Final				25%