

Programming for Data Science - Fall 2026

1. Course Information

Contact Information

- **Instructor:** Dr. Terry Griffin
- **Office:** Bolin Hall 124-F
- **Office Hours:** MW: 2:30 p.m. – 4:00 p.m.; TTh: 11:00 a.m. – 12:00 p.m.
- **Office Phone:** (940) 397-4439
- **Email:** terry.griffin@msutexas.edu

Important Dates, Times, and Location

- **Course:** CMPS 3603 — Programming for Data Science
- **Days:** TBD
- **Time:** TBD
- **Location:** TBD
- **Semester Start:** Monday, August 24
- **Semester End:** Friday, December 4
- **Labor Day:** Monday, September 7
- **Thanksgiving Holiday:** Wednesday, November 25 – Sunday, November 29
- **Last Day for "W":** Monday, November 23
- **Last Assignment Due:** Monday, November 30
- **Final Exam:** TBD (university final-exam schedule)

Course Communication

- **Zoom:** <https://msutexas-edu.zoom.us/j/9403974439>
- **GitHub:** This repository (fork it as described in [Start Here](#))

Course announcements, assignment information, questions, and class discussion may be distributed through the course communication platform and GitHub.

Textbook and Course Materials

Required Textbooks: Free and open source. Copies are included in this repository.

- **[A Whirlwind Tour of Python](#)** by Jake VanderPlas

- **[Python Data Science Handbook](#)** by Jake VanderPlas

Students will need a computer that can run Python 3.13, VS Code, Jupyter notebooks, and Git. The course runs entirely on a **local VS Code + venv + git** setup, not Colab or Codespaces. Course dependencies (numpy, pandas, matplotlib, seaborn, scikit-learn, jupyter) are listed in [01-StartHere/requirements.txt](#) .

Course Repository

Folder	Contents
01-StartHere/	Setup guide: GitHub, Git, Python, VS Code, venv, submitting
02-Assignments/	Numbered course modules (notebooks, glossary, quiz, worksheet)
03-Completed/	Where your finished notebooks go for grading
04-Resources/	Textbooks, shared datasets, and archived modules

New to the course? Work through [01-StartHere/](#) before starting an assignment:

Part	Covers
0	Checking what's already installed (Git, Python, VS Code, Conda)
1	GitHub account + git identity
2	Installing/repairing Git, Python 3.13, and VS Code
3	Creating and connecting your course repository (<code>upstream</code> / <code>origin</code>)
4	Python virtual environment + <code>requirements.txt</code>
5	Using Jupyter notebooks in VS Code
6	Committing and submitting your work
7	Maintaining your repo and troubleshooting git
8	Quick reference for the whole workflow

Prerequisites and Expected Background

This course is beginner-friendly. No prior Python experience is assumed.

Students should be comfortable:

- using a computer, a file system, and a web browser;
- installing software and following written setup instructions;
- reading technical documentation; and
- basic algebra.

2. Course Overview

Course Description

This course introduces programming in Python with an emphasis on working with data. Students start with the core language: values, text, containers, functions, and loops. They then move into the tools data scientists use every day: Jupyter notebooks, Pandas, Matplotlib, and Seaborn.

By the end of the semester, students will load real datasets, clean and reshape them, summarize them, and communicate what they find with visualizations.

Course Approach

The course is **project-focused** and **notebook-driven**. Each module is a set of Jupyter notebooks that explain a concept, show it working, and then ask the student to try it.

Modules 01–04 teach the Python **language**. Module 05 switches to the **tools**. Modules 06 and 07 apply both to real data.

01 → 02 → 03	Core Python: values, text, and containers
↓	
03b	Bridge: choosing containers and breaking problems into functions
↓	
04	The rest of functions and loops, plus reading files
↓	
05	Tools: Jupyter, help, Markdown, files, Pandas, NumPy, plotting, timing
↓	
06	Working with data: Pandas Series/DataFrames, cleaning, grouping
↓	
07	Describing and visualizing data: statistics, Matplotlib, Seaborn

The objective is not to memorize syntax. The objective is to be able to take a question and a dataset and produce a clear, reproducible answer.

3. Learning Objectives

By the end of the course, students should be able to:

1. Set up and maintain a local Python development environment with VS Code, a virtual environment, and Git.
2. Use Python's scalar types, operators, and control flow to express decisions.
3. Manipulate text with string indexing, slicing, methods, and f-strings.
4. Choose among lists, tuples, and dictionaries to represent data, including nested structures and JSON.
5. Break problems into functions that take inputs and return results.
6. Use loops and common patterns (accumulate, count, filter) to process collections and files.
7. Handle errors with `try / except` and read files safely with `with`.
8. Work productively in Jupyter: magics, Markdown, getting help, and timing code.

9. Load, inspect, index, clean, sort, filter, and group tabular data with Pandas.
10. Compute and interpret summary statistics.
11. Create and customize visualizations with Matplotlib and Seaborn.
12. Produce reproducible notebooks that run cleanly from top to bottom.
13. Manage and submit work through a professional GitHub repository.

4. Course Content

Course Modules

Each module folder contains numbered notebooks, a `README.md` , `glossary.md` , `quiz.md` , and `worksheet.md` . Most notebooks end with an optional 🔥 Challenge section.

#	Module	Topics
01	<u>Scalar Types and Control Flow</u>	<code>int</code> , <code>float</code> , <code>str</code> , <code>bool</code> , arithmetic, type casting, comparisons, <code>if / elif / else</code> , logical operators
02	<u>Strings and Text</u>	Strings as sequences, indexing and slicing, immutability, string methods, f-strings and format specs
03	<u>Python Containers</u>	Lists (including 2D lists), tuples, dictionaries, lists of dictionaries as table rows, JSON/GeoJSON
03b	<u>Containers to Functions</u>	Choosing containers, accumulate/count/filter patterns, why functions exist, <code>print()</code> vs. <code>return</code> , pipelines
04	<u>More Functions and Loops</u>	Default and keyword arguments, returning tuples, scope, <code>while / break / continue</code> , <code>with</code> , <code>try / except</code>
05	<u>Foundations</u>	Magics, notebook workflow, getting help, Markdown, files and paths, data I/O, plotting basics, timing
06	<u>Working with Data</u>	Pandas <code>Series / DataFrame</code> , loading and exploring, indexing, cleaning, column operations, sorting, grouping
07	Describing and Visualizing Data <i>(coming soon)</i>	Types of data, summary statistics, Matplotlib basics, Seaborn basics, visualization mini-lab

Major Course Themes

Core Python

- values, types, and operators;
- conditionals and boolean logic;
- strings and text processing;
- lists, tuples, and dictionaries;

- nested data and JSON;
- functions, arguments, return values, and scope;
- `for` and `while` loops; and
- reading files and handling errors.

The Data Science Toolkit

- Jupyter notebooks and IPython magics;
- Markdown documentation;
- files, paths, and data input/output;
- NumPy basics; and
- timing and performance.

Working with Data

- Pandas `Series` and `DataFrame` ;
- loading and exploring real datasets;
- indexing and selection;
- cleaning messy data;
- creating and transforming columns;
- sorting and filtering; and
- grouping and combining.

Describing and Visualizing Data

- types of data;
- summary statistics;
- Matplotlib figures and customization;
- Seaborn statistical plots; and
- telling a clear story with a chart.

Supporting Topics

The following topics are introduced as needed:

- Git and GitHub workflow;
 - virtual environments and package management;
 - debugging;
 - reproducibility (“Restart & Run All”); and
 - reading documentation.
-

5. Coursework and Assessment

Module Assignments

Each module is an assignment. Students complete the module’s notebooks, including the 🏆 Practice problems, and submit them through GitHub.

Notebooks must **Restart & Run All** without errors. Notebooks that create files should submit the matching `NN-OutputFiles/` folder as well.

Module Quizzes

Each module has a quiz covering its notebooks, glossary, and worksheet. Use the module’s `quiz.md` and `worksheet.md` to prepare.

Final Project

The final project asks students to take a real dataset, clean it, analyze it, and present their findings with visualizations in a well-documented notebook. Project requirements will be provided separately.

Final Examination

The final examination will be comprehensive. It will emphasize reading and reasoning about Python and Pandas code, choosing appropriate data structures, and interpreting data summaries and visualizations.

GitHub Portfolio

Students will maintain a GitHub repository containing their work from the course. The portfolio should be organized, readable, and reproducible, and should demonstrate both completed work and professional organization.

6. Grading

Grade Distribution

Category	Weight	Notes
Module Assignments	40%	7 assignments, equally weighted
Module Quizzes	15%	One per module, due with the module
Final Project	15%	Due Monday, November 30
Final Examination	15%	University final-exam schedule
GitHub Portfolio	10%	Resume builder
Participation	5%	Just show up
Total	100%	

Grade Scale

Grade	Percentage
A	90–100
B	80–89
C	70–79
D	60–69
F	Below 60

Grading Notes

Notebooks may be evaluated for:

- correctness;
- completeness, including Practice problems;
- running cleanly from top to bottom;
- readability and Markdown documentation; and
- the student’s ability to explain the submitted work.

🔥 Challenge sections are optional and are not required for full credit.

7. Assignment Calendar

All assignments are due by **11:59 p.m.** on the date listed. Submit by copying finished notebooks into 03–Completed/ , then committing and pushing (see Part 6).

Due Date	Assignment	Quiz
Wednesday, October 14	01 — Scalar Types and Control Flow	Quiz 01
	02 — Strings and Text	Quiz 02
Wednesday, October 21	03 — Python Containers	Quiz 03
	03b — Containers to Functions	Quiz 03b
Wednesday, October 28	04 — More Functions and Loops	Quiz 04
Wednesday, November 4	05 — Foundations	Quiz 05
Wednesday, November 11	06 — Working with Data	Quiz 06
Wednesday, November 18	07 — Describing and Visualizing Data	Quiz 07
<i>November 25 – 29</i>	<i>Thanksgiving Holiday — nothing due</i>	
Monday, November 30	Final Project + GitHub Portfolio	
TBD	Final Examination	

Dates may be adjusted based on class progress. Any changes will be announced in class and through the course communication platform.

8. Common Course Policies

Participation

This course depends on discussion, experimentation, troubleshooting, and asking questions. Meaningful participation includes attending class, asking and answering questions, helping classmates with conceptual problems, and sharing useful discoveries with the class.

Questions are especially valuable. A question asked by one student frequently identifies something that should be clarified for the entire class.

Course Delivery

Course topics may not always be presented in a fixed sequence. Instruction may be adjusted based on class progress, student questions, and topics that arise naturally while solving problems. Core course objectives will remain consistent regardless of the exact order in which material is presented.

Assignment and Submission Requirements

Unless otherwise stated, assignments should:

- satisfy the stated requirements;

- contain all required files, including any `NN-OutputFiles/` folders;
- run cleanly with **Restart & Run All**; and
- be submitted through the course GitHub workflow.

Students are responsible for confirming that work has been successfully pushed to GitHub. Work submitted in the wrong location or format may be treated as not submitted.

GitHub and Repository Requirements

Repositories should be organized, readable, and reproducible. Passwords, private keys, API keys, tokens, credentials, or other sensitive information must not be committed to GitHub.

Understanding and Oral Defense of Submitted Work

Students may be asked to explain, modify, or extend submitted notebooks. Inability to explain submitted work may result in deductions and may result in further review for academic misconduct.

Use of Large Language Models and AI-Assisted Tools

Large language models and AI-assisted tools may be useful for explanation, brainstorming, debugging, and learning unfamiliar syntax. These tools should support learning rather than replace it.

Students remain responsible for understanding, verifying, testing, and being able to explain all submitted work. Submitting generated work that a student does not understand does not demonstrate mastery of course material. Individual assignments may impose additional restrictions.

Attendance

Regular attendance is strongly encouraged and attendance may be recorded. Students are responsible for announcements, material, assignments, and deadlines missed because of an absence.

Classroom Conduct

Students are expected to maintain a classroom environment conducive to learning. Electronic devices may be used when appropriate to course activities. Devices used in a distracting or disruptive manner may be required to be put away.

Computer and Internet Requirements

Students must have reasonable access to a computer and the Internet. Personal computer problems generally do not constitute grounds for extending deadlines. Push work to GitHub regularly, avoid waiting until immediately before a deadline, and notify the instructor promptly of significant technical problems. Campus computing resources are available when personal equipment is unavailable.

Late Work

Late work may be accepted unless an assignment specifies otherwise. Unless otherwise stated:

- the initial late penalty is **15 points**;
- an additional **5 points** may be deducted for each subsequent class period;
- penalties may continue until the assignment has lost up to 50 points; and
- extremely late work may be refused at the instructor's discretion.

Work due on **Monday, November 30** has a firm deadline and will not be accepted late.

Missed Exams and Quizzes

Students who miss an examination or quiz must contact the instructor as soon as possible. Planned absences should be arranged in advance. Make-up work, when permitted, is at the instructor's discretion. An unexcused, undocumented missed examination or quiz may receive a grade of zero.

Final Examination Scheduling

The final examination will be administered at the officially scheduled university date and time. Students should plan travel and other personal activities around the university final-examination schedule. The final examination will not normally be administered early for personal convenience.

Testing Procedures

Unless otherwise announced, bags, books, and electronic devices will be stored in an area designated by the instructor, and a student who leaves the examination room may be required to submit the examination before leaving.

Academic Collaboration

Students are encouraged to discuss course concepts and help one another. Collaboration becomes academic misconduct when a student submits another person's work as their own or provides work for another student to submit. Allowing another student to copy submitted work may constitute collusion.

Recording of Classes

Students must obtain permission before recording class activities unless an approved accommodation provides otherwise. Course materials may not be publicly distributed or posted without permission.

9. Department Resources and Policies

Computer Science Tutoring

Computer Science tutoring is available in **Bolin Room 119** and through the **Office of Tutoring and Academic Support Programs (TASP)** in Moffett Library. Tutors may assist with course concepts and debugging

strategies but may not complete assignments for students.

Department Testing Policy

Department and university testing procedures apply to course examinations. Students should follow instructor directions for each examination.

Department Programming Assignment Policy

Students are responsible for understanding programming assignments submitted for credit. Tests and quizzes may include material from out-of-class assignments. Students may be asked to orally answer questions about submitted work.

Department Academic Misconduct Procedures

Cheating, collusion, plagiarism, and other forms of academic misconduct are subject to Department of Computer Science and university policies. Penalties may include a non-replaceable zero, a reduced course grade, course failure for repeated offenses, and reporting to department and university officials.

10. University Policies and Resources

Academic Misconduct

Students are responsible for reviewing the current **MSU Student Handbook, Appendix E: Academic Misconduct Policy & Procedures**.

https://msutexas.edu/student-life/_assets/files/handbook.pdf

Students Requiring Accommodations

Students who require academic accommodations because of a disability should contact the Disability Support Office in Clark Student Center 168, **(940) 397-4140**, and provide the required documentation as early as possible.

Midterm Progress Reports

Midterm progress information will be provided through the MSU Portal. Midterm grades are not reported on the permanent transcript. Students earning below a C at midterm are strongly encouraged to meet with the instructor and use available tutoring resources.

Campus Carry

Students should consult the current MSU Campus Carry Policy:

Tobacco Policy

Midwestern State University prohibits the use of tobacco products in university facilities and other locations covered by university policy.

Moffett Library

Moffett Library provides research assistance, databases, study spaces, and remote access to electronic resources.

Student Technical Support

Students having trouble with university technology should use the university's student technical-support resources and notify the instructor promptly when a system problem affects course work.

Tutoring and Academic Support

Students are encouraged to use university tutoring and academic-support programs. Seeking help early is strongly preferable to waiting until right before a deadline.

University policies and services may change. Current official university policy takes precedence over this summary.