

Advanced Programming with Python

CS 122

Fall 2026 Section 01 In Person 3 Unit(s) 08/19/2026 to 12/07/2026 Modified 08/26/2026

Contact Information

My name is Mike Wood and I look forward to working with you as we learn programming techniques in Python in CS 122!

Instructor: Dr. Mike Wood

Email: mike.wood@sjsu.edu

I welcome you to contact me outside of class either by email or by joining me in office hours. If your schedule prevents you from joining my scheduled office hours, please get in touch and I will be happy to schedule a time slot to meet with you.

Office Hours

Tuesday, 1:00 PM to 2:00 PM, Virtual

Virtual office hours can be accessed via the Zoom link on Canvas

Course Description and Requisites

Advanced features of the Python programming language with emphasis on programming practice. Course involves substantial programming projects in Python.

Prerequisite(s): CS 146 (with a grade of "C-" or better) or instructor consent. Computer Science, Data Science, Computer Science and Linguistics, Applied and Computational Math, or Software Engineering majors only.

Grading: Letter Graded

* Classroom Protocols

Classroom Learning Environment:

This course will follow a hands-on learning approach where we will work through coding exercises together in class. Please come to class with a charged laptop ready to dive into some code!

Code of conduct:

Short version: In this course, I aim to foster a positive learning environment - no form of harassment will be tolerated, including verbal comments and images that exclude people based on gender, socio-economic status, or appearance.

The full code of conduct is provided on the Canvas course space for this course.

Plagiarism and cheating

Just like a written essay, using somebody's computer code without proper acknowledgement is considered plagiarism. Homework problems should be based entirely on students' own work. Students can (and are encouraged to) discuss general coding techniques and problem solving strategies for homework problems but this should never include copying (whether by typing, file transfer or cutting and pasting), looking at somebody else's code on their computer to get help, or allowing copying to occur. Students found violating this policy once will receive zero credit for those problems. Continued violations will result in disciplinary action. If you have any questions about this policy, please don't hesitate to ask for clarification.

AI Policy

Short Version

The AI policy for this course can be summarized in two stages:

First half: Learn to program without AI.

The goal of the first part of this course is to build fluency in Python so that you can reason about code yourself and critically evaluate what an LLM produces. To support this goal, class work and home work assignments will be completed without the assistance of AI.

Second half: Leverage AI to generate more sophisticated programs

The goal of the second part of this course is to explore different Python packages and capabilities. In this part, assignments will use AI as a powerful development tool but students remain responsible for understanding, debugging, evaluating, and improving everything submitted as assignments.

Long Version

Generative AI is rapidly changing how programmers write, debug, and understand code. In this course, students will learn both how to program first without relying on AI and then how to use AI tools effectively and responsibly. The AI policy therefore changes during the semester.

First Half of the Course: Building Python Skills

The goal of the first half of the course is to develop a strong understanding of the nuances of Python's fundamental building blocks. Students should be able to read, write, and reason about Python code on their own before relying on AI assistance. For regular homework assignments, AI use is not permitted.

Students are expected to complete homework without using generative AI tools i.e. without GitHub Copilot and similar AI-assisted coding features, and without ChatGPT, Claude, Gemini, or other LLMs. Traditional documentation and course resources should still be consulted. The purpose of this restriction is not to pretend that AI tools do not exist. It is to make sure that students develop the programming knowledge needed to recognize when AI-generated code is correct, incorrect, elegant, inefficient, or simply strange.

While students are expected to complete assignments without AI, each assignment in the first half of the class will include a section in which student are specifically asked to use an LLM to generate a solution. In these exercises, the goal is to compare human- and machine-generated solutions and analyze differences.

The midterm will be handwritten and closed-note i.e. students will need to demonstrate that an understanding of Python code independently. This is an additional incentive to use the first half of the semester to build fluency in Python rather than outsourcing this practice to an LLM.

Second Half of the Course: Use Python to Do Cool Things

In the second half of the course, the emphasis shifts - instead of concentrating primarily on the mechanics of the Python language, Python packages will be used to solve more substantial problems and build bigger programs. At this point, students are welcome to use their favorite LLM or coding tools to help with assignments and the class project unless an assignment states otherwise. However, permission to use AI does not mean that responsibility for your code has been transferred to the AI. Students must be able to explain their code and debug issues.

Program Information

Diversity Statement - At SJSU, it is important to create a safe learning environment where we can explore, learn, and grow together. We strive to build a diverse, equitable, inclusive culture that values, encourages, and supports students from all backgrounds and experiences.

Course Learning Outcomes (CLOs)

Upon successful completion of this course, students will be able to:

1. Design, implement and test readable, efficient programs that take advantage of Python built-in capabilities and follow Python best practices.
2. Understand implementation differences and performance tradeoffs associated with various Python data structures.
3. Develop Python applications using the modules and packages available in the Python standard library.
4. Develop Python applications using third party libraries.
5. Design, implement and test Python programs that include a graphical user interface, data analysis and visualization, web data extraction and web applications.

Course Materials

This course will utilize The Quick Python Book by Naomi Cedar (3rd Edition, ISBN 9781617294037).

The Quick Python Book

Author: Naomi Cedar

Edition: 3rd

ISBN: 9781617294037

The Quick Python Book is also available at the MLK Library, including electronic access options.

Grading Information

Category	Percent of Total Grade
Reading Quizzes	10
Homework	30
Midterm	20
Project	40

University Policies

Per [University Policy S16-9 \(PDF\)](http://www.sjsu.edu/senate/docs/S16-9.pdf) (<http://www.sjsu.edu/senate/docs/S16-9.pdf>), relevant university policy concerning all courses, such as student responsibilities, academic integrity, accommodations, dropping and adding, consent for recording of class, etc. and available student services (e.g. learning assistance, counseling, and other resources) are listed on the [Syllabus Information](https://www.sjsu.edu/curriculum/courses/syllabus-info.php) (<https://www.sjsu.edu/curriculum/courses/syllabus-info.php>) web page. Make sure to visit this page to review and be aware of these university policies and resources.

Course Schedule

This course will meet twice per week during the Spring semester as follows:

- Tuesdays, virtually on Zoom, 3:00-4:15 pm
- Thursdays, in-person in DH 135, 3:00-4:15 pm

Week	Day	Date	Lecture	Pre-Class Reading	Homework
1	Thurs	8/20	Course Logistics & Installation	Ch. 1	
2	Tues	8/25	Python Basics	Ch. 4	HW 1 Due 8/24
2	Thurs	8/27	Sequence Data Types (Lists, Tuples, and Sets); Strings Part 1	Ch. 5, 6	
3	Tues	9/1	Strings Part 2; Dictionaries	Ch 7	HW 2 Due 8/31
3	Thurs	9/3	Control Flow	Ch. 8	
4	Tues	9/8	Functions	Ch 9	
4	Thurs	9/10	Modules, Scoping, Programs	Ch. 10, 11	HW 3 Due 9/9
5	Tues	9/15	Using the Filesystem	Ch. 12	
5	Thurs	9/17	Files I/O	Ch. 13	HW 4 Due 9/16
6	Tues	9/22	Exceptions	Ch. 14	
6	Thurs	9/24	Python Classes	Ch. 15	HW 5 Due 9/23
7	Tues	9/29	Objects	Ch. 17	
7	Thurs	10/1	Packages, Libraries	Ch. 18, 19	
8	Tues	10/6	File Wrangling		HW 6 Due 10/5
8	Thurs	10/8	Processing Data Files	Ch. 21	
9	Tues	10/13	Github Final Project Formulation		
9	Thurs	10/15	Midterm		
10	Tues	10/20	Scraping the Web (Pt. 1)	Ch. 22	HW 7 Due 10/19
10	Thurs	10/22	Scraping the Web (Pt. 2)		
11	Tues	10/27	GUI programming (tkinter, Pt. 1)	See Canvas	HW 8 Due 10/26
11	Thurs	10/29	GUI programming (tkinter, Pt. 2)		
12	Tues	11/3	Web Development (flask, Pt. 1)	See Canvas	HW 9 Due 11/2
12	Thurs	11/5	Web Development (flask, Pt. 2)		

13	Tues	11/10	Data Analysis with numpy	See Canvas	HW 10 Due 11/9
13	Thurs	11/12	Data Analysis with pandas	Ch. 24	
14	Tues	11/17	Visualization with matplotlib		
14	Thurs	11/19	Visualization with pyplot		
15	Tues	11/24	Making movies		HW 11 Due 11/23
15	Thurs	11/26	Thanksgiving (no class)		
16	Tues	12/1	Final Project Check-Ins		
16	Thurs	12/3	Final Project Check-Ins		
		12/10	Final Projects		Final Projects Due 12/10 at 1 pm