

CSA 1320: Computer Science II

Department of Computer Science

Fall Semester 2026

Faculty Information

Meg Stuart

Email: mstuart@sulross.edu

Office Hours: Wednesday 2-4 pm, online or in-person, please email ahead of time

Course Description

This course introduces students to the fundamental concepts and techniques of object-oriented programming (OOP). Students will learn to design and build programs using objects, classes, constructors, methods, object inheritance, and polymorphism from a practical, hands-on approach. The course also covers exception handling, file input/output, and testing and debugging of object-oriented code. Throughout the semester, topics are connected to current applications in artificial intelligence (AI), machine learning (ML), and natural language processing (NLP), showing students how the object-oriented style they are learning is the same style used to build modern AI systems.

Prerequisite: CSA 1309 (Computer Science I), or equivalent under prior course numbering.

Programming Language: Python (primary). Examples in other languages may be shown for comparison.

Course Materials

Textbook (Recommended):

- Eric Matthes, Python Crash Course, 3rd Edition (No Starch Press). Chapters on classes, files, exceptions, and testing.

Program Student Learning Outcomes

Upon completion, students will be able to:

- Explain the core concepts of object-oriented programming: objects, classes, constructors, methods, inheritance, and polymorphism.
- Design and implement classes that model real-world entities and relationships.
- Apply inheritance and polymorphism to build extensible class hierarchies.
- Use composition to build programs from cooperating objects.
- Handle errors gracefully using exception handling and input validation.
- Read and write data using text, CSV, and JSON files.
- Test and debug object-oriented programs systematically.

Important Dates

- Aug 25 – First Day (Introductions only)
- Sep 1 – Classes start (Week 2)

- Nov 27–28 – Thanksgiving Break (No Class)
- Dec 3 – Last Day to Submit All Work & Capstone Project Due
- Dec 10 – Last Day of Class (Reserved for grading, no new work)

Course Student Learning Outcomes

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

- Write correct, well-organized object-oriented programs in Python.
- Choose appropriately between composition and inheritance when designing programs.
- Build programs that persist and retrieve data through file input/output.
- Write unit tests for classes and use debugging tools to locate and fix defects.
- Recognize how object-oriented design underpins AI, ML, and NLP software libraries.
- Present solutions and design reasoning clearly in both code and written form.

Marketable Skills

Students will develop:

- **Problem-Solving & Critical Thinking** – Modeling real-world problems as interacting objects.
- **Technical Literacy** – Object-oriented programming, testing, and debugging.
- **Analytical Skills** – Evaluating design alternatives (composition vs. inheritance).
- **Communication Skills** – Explaining design choices in writing and orally.

Course Quizzes, Assignments, and Grading

Assignment Type	% of Final Grade	Description
Weekly Assignments	70%	Applied coding and reflection exercises that build each week (approximately 14 assignments, ~5% each).
Final Capstone Project	30%	A comprehensive object-oriented programming project integrating classes, inheritance, exception handling, file I/O, and testing.
Total	100%	

Late Assignment Statement:

Assignments are due by 11:59 PM on the posted due date. Late submissions will be accepted up to 48 hours past the deadline with a 10% deduction per day. Work submitted more than 48 hours after the due date will not be accepted unless prior arrangements have been made with the instructor for extenuating circumstances. Participation in discussions must follow module timelines; late posts may not receive credit.

Grading Rubrics

Weekly Assignments (70% of grade) — Rubric (approx. 5% each):

- Understanding of Concept (clear, correct, thoughtful)
- Application & Examples (program works, meaningful examples)
- Clarity & Organization (code is readable, structured)
- Completion (all requirements met)

Capstone Project (30% of grade) — Rubric:

- Integration of Course Content (uses classes, inheritance, exceptions, file I/O, and testing)
- Depth & Creativity (originality and effort)
- Correctness & Efficiency (program runs correctly and is well designed)
- Clarity & Professionalism (report/presentation clear and well organized)

Course Schedule

Week	Topic	Notes
1	Course Introduction	From Computer Science I to II: where we are headed. Overview of syllabus, expectations, and tools. Demo: a procedural program vs. the same program written with objects. AI connection: modern AI libraries (scikit-learn, PyTorch, TensorFlow) are built entirely from classes and objects — this course teaches you to read and write that style of code. Assignment 1: Hello World
2	CS I Review & Program Design	Review of functions, lists, dictionaries, loops, and file basics from Computer Science I. Decomposing a problem into functions. Assignment 2: Refactoring messy code
3	Introduction to Object-Oriented Programming	What are objects and classes? Objects as state + behavior; the OOP mindset vs. procedural programming. Real-world modeling examples. AI connection: a trained ML model is an object — data (weights) plus behavior (predict). Assignment 3: Refactoring to object oriented structure
4	Classes & Constructors	Defining classes; the constructor (<code>__init__</code>); instance attributes; creating and using multiple objects. Mini exercises: Student class, BankAccount class. NLP connection: building a simple Document class that stores text and metadata.
5	Methods & Encapsulation	Instance methods; self; getters/setters and properties; information hiding and why it matters. AI connection: why ML libraries hide internal state behind clean method interfaces.
6	Objects Working Together (Composition)	Classes containing other objects; modeling relationships (a Course has Students; a Library has Books). Assignment: mini text-analysis tool — a Corpus class composed of Document objects that counts word frequencies (a first step toward NLP).
7	Inheritance I	Base and derived classes; inheriting attributes and methods; extending behavior; method overriding. AI connection: in deep-learning frameworks,

Week	Topic	Notes
		custom models are written by inheriting from a base class (e.g., extending a framework's Model or Module class).
8	Inheritance II & Polymorphism	Multi-level hierarchies; polymorphism — one interface, many behaviors; duck typing in Python. AI connection: every scikit-learn model shares the same fit()/predict() interface — polymorphism is what makes swapping one ML model for another a one-line change.
9	Exception Handling & Defensive Programming	Errors vs. exceptions; try/except/finally; raising exceptions; validating input. AI connection: robust data pipelines — handling missing or malformed data before it reaches a model.
10	File I/O & Working with Data	Reading and writing text, CSV, and JSON files; parsing structured data into objects. NLP connection: loading a small text corpus from files and computing simple statistics (word counts, vocabulary size).
11	Testing & Debugging Object-Oriented Code	Unit testing with unittest/pytest; writing testable classes; using the debugger; reading tracebacks. AI connection: why testing matters even more in ML systems, where silent data errors produce wrong answers instead of crashes.
12	Class Design & UML Basics	Designing before coding: identifying classes, responsibilities, and relationships; simple UML class diagrams; introduction to design principles (single responsibility, favoring composition). Capstone proposal due.
13	—	Thanksgiving Break (No Class).
14	Putting It All Together	Larger worked example integrating classes, inheritance, exceptions, file I/O, and testing. Capstone work time and instructor check-ins. AI connection: walkthrough of how a simple sentiment-analysis tool would be organized as a set of cooperating classes — previewing ideas students will formalize in Data Structures and later AI coursework.
15	Capstone Project & Wrap-Up	Capstone submission and presentations. Options: library management system (composition + file I/O); text-analysis toolkit (Corpus/Document classes + word statistics); simple game with a character class hierarchy (inheritance + polymorphism); gradebook application (classes + exceptions + CSV persistence).
16	Grading Week	Reserved for project grading and feedback; no new work.

ADA Statement

SRSU Accessibility Services. Sul Ross State University (SRSU) is committed to equal access in compliance with the Americans with Disabilities Act of 1973. It is SRSU policy to provide reasonable accommodations to students with documented disabilities. It is the student's responsibility to initiate a request each semester for each class. Students seeking accessibility/accommodations services must contact Mrs. Mary Schwartz Grisham, LPC, SRSU's Accessibility Services Director, or Ronnie Harris, LPC, Counselor, at 432-837-8203 or email mschwartz@sulross.edu or ronnie.harris@sulross.edu. RGC students can also contact Alejandra Valdez at 830-758-5006 or email alejandra.valdez@sulross.edu. Our office is located

on the first floor of Ferguson Hall, room 112, and our mailing address is P.O. Box C122, Sul Ross State University, Alpine, Texas, 79832.

Student Responsibilities Statement

All full-time and part-time students are responsible for familiarizing themselves with the Student Handbook and the Undergraduate & Graduate Catalog and for abiding by the University rules and regulations. Additionally, students are responsible for checking their Sul Ross email as an official form of communication from the university. Every student is expected to obey all federal, state, and local laws and is expected to become familiar with the requirements of such laws.

SRSU Distance Education Statement

Students enrolled in distance education courses have equal access to the university's academic support services, such as library resources, online databases, and instructional technology support. For more information about accessing these resources, visit the SRSU website.

Students should correspond using Sul Ross email accounts and submit online assignments through Blackboard, which requires a secure login. Students enrolled in distance education courses at Sul Ross are expected to adhere to all policies pertaining to academic honesty and appropriate student conduct, as described in the student handbook. Students in web-based courses must maintain appropriate equipment and software, according to the needs and requirements of the course, as outlined on the SRSU website. Directions for filing a student complaint are located in the student handbook.

Counseling

Sul Ross has partnered with TimelyCare, where all SR students have access to nine free counseling sessions. You can learn more about this 24/7/365 support by visiting TimelyCare/SRSU. The SR Counseling and Accessibility Services office will continue to offer in-person counseling in Ferguson Hall room 112 (Alpine campus), and telehealth Zoom sessions for remote students and RGC students.

Libraries

The Bryan Wildenthal Memorial Library and Archives of the Big Bend in Alpine offer FREE resources and services to the entire SRSU community. Access and borrow books, articles, and more by visiting the library's website, library.sulross.edu. Off-campus access requires logging in with your LoboID and password. Librarians are a tremendous resource for your coursework and can be reached in person, by email (srsulibrary@sulross.edu), or by phone (432-837-8123).

No matter where you are based, public libraries and many academic and special libraries welcome the general public into their spaces for study. SRSU TexShare Cardholders can access additional services and resources at various libraries across Texas. Learn more about the TexShare program by visiting library.sulross.edu/find-and-borrow/texshare/ or ask a librarian by emailing srsulibrary@sulross.edu.

Mike Fernandez, SRSU Librarian, is based in Eagle Pass (Building D-129) to offer specialized library services to students, faculty, and staff. Utilize free services such as InterLibrary Loan (ILL), ScanIt, and Direct Mail to get materials delivered to you at home or via email.

Academic Integrity

Students in this class are expected to demonstrate scholarly behavior and academic honesty in the use of intellectual property. Students should submit work that is their own and avoid the temptation to

engage in behaviors that violate academic integrity, such as turning in work as original that was used in whole or part for another course and/or professor; turning in another person's work as one's own; copying from professional works or internet sites without citation; or collaborating on a course assignment, examination, or quiz when collaboration is forbidden. Students should also avoid using open AI sources unless permission is expressly given for an assignment or course. Violations of academic integrity can result in failing assignments, failing a class, and/or more serious university consequences. These behaviors also erode the value of college degrees and higher education overall.

Classroom Climate of Respect

Importantly, this class will foster free expression, critical investigation, and the open discussion of ideas. This means that all of us must help create and sustain an atmosphere of tolerance, civility, and respect for the viewpoints of others. Similarly, we must all learn how to probe, oppose, and disagree without resorting to tactics of intimidation, harassment, or personal attack. No one is entitled to harass, belittle, or discriminate against another on the basis of race, religion, ethnicity, age, gender, national origin, or sexual preference. Still, we will not be silenced by the difficulty of fruitfully discussing politically sensitive issues.

Supportive Statement

I aim to create a learning environment for my students that supports various perspectives and experiences. I understand that economic disparity, health concerns, or unexpected life events may impact the conditions necessary for you to succeed. My commitment is to be there for you and help you meet the learning objectives of this course. I do this to demonstrate my commitment to you and to the mission of Sul Ross State University to create a supportive environment and care for the whole student as part of the Sul Ross Familia. If you feel like your performance in the class is being impacted by your experiences outside of class, please don't hesitate to come and talk with me. I want to be a resource for you.

Use of Generative Artificial Intelligence (AI)

University Statement (required):

The University does not recommend or endorse any specific AI tools or resources. Students should be aware that many generative AI tools (e.g., ChatGPT, Google Gemini, Microsoft Copilot) store user input and may use this data to train future models. For this reason, students should never upload or share personal, confidential, or identifiable information—such as names, ID numbers, health data, or assignment submissions containing such details—into any generative AI platform. When using AI tools, students should verify whether the tool complies with student privacy standards as indicated by the University. Faculty may recommend specific tools that better align with institutional data privacy policies, but ultimate responsibility for data protection rests with users. Students are encouraged to use faculty-recommended platforms when engaging in coursework involving generative AI. The University is not liable for any adverse experience or impact when students interact with these tools.

Course Policy — Use of Generative AI to write code is not permitted, other use may be allowed with acknowledgment:

Generative AI tools may be useful for brainstorming ideas, exploring possible approaches to problems, and receiving feedback on your own work. In this course, students shall only use generative AI to brainstorm ideas and never to generate text or code for assignments. Students shall give credit to AI tools whenever they are used. When using AI tools on assignments, add a brief appendix identifying (a)

which AI tools were used, (b) how they were used (e.g., to generate ideas, debug code, clarify a concept), and (c) why they were used. All submitted code and written work must be your own; submitting AI-generated work as your own, or failing to acknowledge the use of generative AI tools, is a violation of university policy and will be treated as academic dishonesty. Students shall not use AI tools during examinations or on specific assignments where their use is expressly prohibited. If you are unsure whether a particular use of AI is acceptable, please ask before submitting.