

Sul Ross State University
Course Syllabus
CSA/CSR 1309: Computer Science I and Lab
Fall 2026

Instructor: Dr. Bushra Sajid

Office Number: ACR 203 C

Email Address: Bushra.sajid@sulross.edu

Office Hours: 3:00 am- 5:00 pm TTR, others by appointment

Time and Place of Class Meetings: Lecture: Tuesday and Thursday: 10:00 am - 10:50 am ACR 203, Lab: 11:00 am - 11:50 am ACR 203

Course Objectives: This course is a disciplined approach to problem-solving with structured techniques and representation of algorithms using pseudo-code and graphical tools. There will be discussions of methods for testing, evaluation, and documentation. Topics include data types; control structures; functions, structures, arrays, and file input/output; the mechanics of running, testing, and debugging programs; introduction to programming; and introduction to the historical and social context of computing. Prerequisite: MATH 1314

Course Materials:

Textbook 1 (Free Online): Allen B. Downey. *Think Python: How to Think Like a Computer Scientist*, 3rd Edition. Green Tea Press, 2024. Free online at: <https://alldowney.github.io/ThinkPython/>

Textbook 2 (Free Online): Charles R. Severance. *Python for Everybody: Exploring Data in Python 3*. Free online through Python for Everybody at: <https://www.py4e.com/book>

Additional readings, programming examples, handouts, interactive exercises, and multimedia resources will be provided throughout the semester.

Required Technology and Software:

Students will need regular access to a computer capable of running Python 3.

Programming activities may use one or more of the following:

- Python 3
- IDLE
- Visual Studio Code
- Google Colab

Additional software or online resources may be introduced during the semester.

Students should maintain backup copies of their programming work.

Marketable Skills:

- Students demonstrate computational thinking and problem-solving skills through the design and implementation of algorithms.
- Students develop fundamental programming skills using data types, control structures, functions, arrays, and file input/output.
- Students apply systematic testing and debugging techniques to identify, analyze, and correct programming errors.
- Students demonstrate technical communication skills through clear pseudocode, program documentation, and well-structured code.
- Code analysis and review.
- Responsible use of artificial intelligence tools.
- Verification of AI-generated technical content.
- Communication of technical solutions.

BS Computer Science Student Learning Outcomes (SLOs):

- Students will be able to comprehend the fundamental concepts of computer science, including the binary number system, string operations, data types, branching, looping, input/output (I/O) operations, basic algorithms, and common data structures.
- Students will be able to demonstrate knowledge of modern computer systems, databases, and networking.
- Students will demonstrate mastery of and the ability to implement modern object-oriented programming methodologies and will be able to design and implement a basic Unified Modeling Language (UML) application design.

Course Learning Objectives: Upon successful completion of the course, students should be able to:

1. Explain the basic process by which a computer executes a program.
2. Translate a real-world problem into a computational problem.
3. Design algorithms using pseudocode and basic flowcharts.
4. Write Python programs using variables, expressions, input, output, and fundamental data types.
5. Apply Boolean logic and conditional statements to decision-making problems.
6. Apply repetition structures including `while` and `for` loops.

7. Develop modular programs using functions, parameters, arguments, and return values.
8. Work with strings and common Python collections including lists, tuples, dictionaries, and sets.
9. Read information from files and write program data to files.
10. Apply exception handling and input validation.
11. Identify and correct syntax, runtime, and logical errors.
12. Explain the behavior of an unfamiliar Python program by tracing its execution.
13. Develop introductory object-oriented programs using classes and objects.
14. Evaluate the correctness of AI-generated Python programs.
15. Use generative AI appropriately for explanation, debugging, testing, documentation, and code review.
16. Design, implement, test, document, and explain a complete Python software project.

Grading Scale: 90-100 A, 80-89 B, 70-79 C, 60-69 D, 59-Below F

Grading Policy:

Quizzes/Homework: 15%

Laboratories: 10%

Midterm Exams: 25%

Final Exam: 40%

The grade weighting will be as follows:

Assignments with their due dates will be posted on Blackboard.

No make-up exams will be given. If there is a valid reason for missing an exam, then the grade for the missed exam will be replaced by the grade on the final exam. Otherwise, a missed exam will be a zero. Exams will be closed notes, closed book, and no calculator will be allowed unless otherwise stated. Any restroom breaks need to be taken before an exam starts. You cannot leave the classroom in the middle of an exam under any circumstances.

Lab time will be used in a variety of ways. It may be used to answer questions we did not get to in class, extra assignments to gain a deeper understanding, or quizzes and exams. If exams are to be given during lab time, I will give ample notice.

Attendance Policy: Students are expected to attend every class. If class must be missed, the student is expected to get the notes from a classmate, and to check with me or on Blackboard for announcements and updated assignments.

Students are expected to arrive to class on time. If a student is perpetually late, they will be asked to not attend class unless they can arrive on time. If tardiness becomes a problem for the class as a whole, people who arrive late will not be permitted to enter the class. If this stricter policy becomes necessary, there will be an announcement made in class.

Cell Phone Policy: Any phone ringing during class will be taken up until the end of class. If a phone rings during a test or quiz, the student will forfeit their right to finish said test or quiz.

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.

Americans With Disabilities Act: 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. To help protect confidentiality, students seeking accessibility/accommodation(s) services must submit the Accommodation Request Form located on our website. Students are welcome to call the Office of Accessibility Services at 432-837-8203 or stop by Ferguson Hall Room 112 if they have any additional questions regarding the process. Our mailing address is PO Box C-122; Alpine, TX; 79832.

Counseling: Sul Ross has partnered with TimelyCare where all SR students will have access to nine free counseling sessions. You can learn more about this 24/7/365 support by visiting [Timelycare/SRSU](https://www.timelycare.com/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.

Library Services: 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 LobolD 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; collaborating on a course assignment, examination, or quiz when collaboration is forbidden. Students should also avoid using OpenAI 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.

Generative Artificial Intelligence (AI): 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.

Student Responsibilities : 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.

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 the recent pandemic, economic disparity, and health concerns, or even 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.

Tutoring: The Lobo Den Tutoring Center offers FREE tutoring support to help you excel in your courses. Whether you need assistance in Writing, Math, Science, or other subjects, we're here to help!

Important Information:

Drop-in and Scheduled Appointments: Flexible options to fit your needs. Hours of Operation: Monday–Friday, 8:00 AM – 5:00 PM. Workshops: Attend our regularly hosted academic workshops on STEM topics and professional development, often in collaboration with specialized faculty. Location: BWML Room 128. Contact Us: For more information or to book an appointment, email tutoring@sulross.edu or call (432) 837-8726. Looking for additional support?

Tutor.com offers FREE 24/7 online tutoring in over 200 subjects, including specialized support for ESL and ELL learners with native Spanish-speaking tutors. Access Tutor.com via Blackboard: Log in to your Blackboard account to get started anytime, anywhere. Take advantage of these valuable resources to boost your confidence and performance in your classes. We look forward to helping you succeed!

Important Dates:

August 25	First class meeting
October 20	Midterm Examination
November 23–27	Thanksgiving Break – no class
December 1	Final project demonstrations and code defense
December 2	End of course instructional period

Additional official university deadlines regarding registration, census dates, withdrawals, and final examinations should be verified against the Fall 2026 Sul Ross State University Academic Calendar.



Tentative Class and Laboratory Schedule:

This syllabus and course schedule are subject to change when necessary to improve instruction, accommodate University scheduling, or respond to the progress and needs of the class.

Changes will be communicated through Blackboard, Sul Ross email, and/or class announcements.

Date	Lecture Topic	Laboratory / Practical Focus
Aug 25	What Is Computer Science? Programming in the Age of AI	First Python program; programming environment; examine and modify simple AI-generated Python code.
Aug 27	Computational Thinking and Algorithms	Transform an everyday problem into inputs, processing, outputs, pseudocode, and Python.
Sept 1	Variables, Data Types, and Memory	Practice <code>int</code> , <code>float</code> , <code>str</code> , <code>bool</code> , assignment, conversion, and program tracing.
Sept 3	Expressions, Operators, Input, and Output	Develop various programs that use user input and produce formatted output.
Sept 8	Boolean Logic and Decision Making	Evaluate Boolean expressions, comparisons, logical operators, and truth conditions.
Sept 10	Conditional Statements: <code>if</code> , <code>elif</code> , and <code>else</code>	Develop a grade calculator, eligibility system, or similar decision-based programs.
Sept 15	Nested Decisions and Input Validation	Develop programs that identify unreasonable or invalid user input.
Sept 17	Introduction to Loops: <code>while</code>	Develop sentinel-controlled programs.
Sept 22	<code>for</code> Loops and <code>range()</code>	Practice counting, totals, averages, patterns, and simple simulations.
Sept 24	Nested Loops and Iterative Problem Solving	Generate tables and patterns and manually trace nested-loop execution.
Sept 29	Functions and Problem Decomposition	Refactor a large sequential program into smaller reusable functions.
Oct 1	Parameters, Arguments, and Return Values	Develop a function-based calculator and practice returning values.
Oct 6	Scope, Local Variables, and Program Design	Trace function execution and debug common scope-related problems.
Oct 8	Strings and Text Processing	Develop a text analyzer using indexing, slicing, string methods, counting, and searching.
Oct 13	Lists and Collections	Develop a student-score or small data-analysis application.
Oct 15	Midterm Review: Tracing, Debugging, and Problem Solving	Repair broken programs and complete an AI-generated buggy-code challenge.
Oct 20	Midterm Examination	
Oct 22	Dictionaries, Tuples, and Sets	Run examples on dictionaries, tuples, and sets.
Oct 27	Files and Persistent Data	Read information from files, process records, and write program results to files.
Oct 29	Exceptions and Defensive Programming	Use <code>try/except</code> , validate input, and improve program resilience.
Nov 3	Debugging and Software Testing	Identify syntax, runtime, and logical errors.
Nov 5	AI-Assisted Programming: Prompt, Inspect, Test, and Verify	AI Code Audit: evaluate an AI-generated solution, identify weaknesses, test it, and produce a corrected version.
Nov 10	Introduction to Objects and Classes	Model a real-world entity using a Python class and create multiple objects.
Nov 12	Constructors, Attributes, and Methods	Develop a small class-based program.
Nov 17	From Problem to Software Project	Develop the final-project proposal.
Nov 19	Project Architecture and Documentation	Create the project repository, README, program structure, functions/classes, and initial working skeleton.
Nov 24	Thanksgiving break	No laboratory.
Nov 26	Thanksgiving break	No laboratory.
Dec 1	Final Project Demonstrations and Code Defense	Students demonstrate their applications, explain selected code, discuss testing, and disclose AI assistance.

The goal of Computer Science I is not simply to learn Python.

The goal is to learn how to take a problem, design an algorithm, implement it, test it, debug it, explain it, and determine whether a computational solution including one produced with artificial intelligence is actually correct.