CSC 141: Computer Science I

Fall 2026 • Section 01 • CRN 34650

Instructor	Dr. Si Chen
Email	schen@wcupa.edu
Office	25 University Ave., Room 131
Department Phone	610-436-6998
Office Hours	Tue/Thu 12:30-2:00 PM (by appointment); Wed 1:00-3:00 PM
Credits	3
Campus	West Chester
Instructional Method	In Person (0-25% Distance Education)
Meeting Time	Tuesday / Thursday, 11:00 AM - 12:15 PM
Location	Mitchell Hall 102
Term Dates	August 24 - December 12, 2026
Course Site	Course website: wcu.ninja/csc141 (slides, schedule, materials). D2L: submissions, grades, announcements.

Syllabus version: August 24, 2026

1. Catalog Information and Course Description

Official Course: CSC 141 - Computer Science I (3 credits)

An introduction to programming using Python. Topics covered include basic program design; program flow including decisions, functions, and loops; command line and file input/output; variables and types; and string and sequence processing.

General Education Attribute: Science Distributive

Delivery: This section is scheduled in person. Distance education components, if any, will not exceed the section's published instructional-method designation.

Prerequisite assumption: No prior programming experience is assumed. The course emphasizes problem decomposition and program design as well as Python syntax.

2. Required and Recommended Materials

Required textbook: None. Required readings, examples, starter files, and practice materials will be posted in D2L. The public course website, wcu.ninja/csc141, posts the schedule, lecture slides, and a browser-based Python playground you can run with no installation; graded submissions and grades remain in D2L.

- Python 3. The course will use the Python 3 version and setup specified in D2L.
- A programming editor/IDE. Visual Studio Code with the Python extension is recommended; IDLE or another instructor-approved Python environment is acceptable.
- Reliable access to D2L and your WCU email account. You are responsible for checking course announcements and messages regularly.
- Regular access to a computer on which you can write, run, test, and submit Python programs. Bring a laptop to class when requested.

3. Course Student Learning Outcomes and Alignment

Upon successful completion of CSC 141, students will be able to:

1. Design algorithmic solutions to introductory computing problems using decomposition, pseudocode, and stepwise refinement. *[Program SLO 1, 2]*
2. Write, run, and debug Python programs using variables, expressions, built-in types, console input/output, and appropriate documentation. *[Program SLO 1, 2, 3]*
3. Use Boolean expressions, selection statements, and iteration to control program flow. *[Program SLO 1, 2]*
4. Define and call functions using parameters and return values, and decompose larger programs into manageable components. *[Program SLO 1, 2]*
5. Process strings and sequences using indexing, slicing, iteration, and common operations. *[Program SLO 1, 2]*
6. Read from and write to files and create programs that interact through command-line/console input and output. *[Program SLO 1, 2]*
7. Test and debug programs systematically by forming hypotheses, examining evidence, and revising solutions. *[Program SLO 1, 2, 3]*

Program SLOs used in this syllabus: SLO 1 - Analyze a complex computing problem; SLO 2 - Design, implement, and evaluate a computing-based solution; SLO 3 - Communicate effectively.

General Education: Science Distributive

General Education Goal	CSC 141 Alignment	Primary Evidence
Goal 1 - Communicate effectively	Students explain program designs through pseudocode, meaningful names, code documentation, and brief written explanations.	Programming assignments; selected exam/quiz responses
Goal 2 - Think critically and analytically	Students analyze requirements, construct algorithms, compare possible solutions, test assumptions, and revise code using evidence from program behavior.	Programming assignments; debugging activities; exams
Goal 3 - Employ quantitative concepts and mathematical methods	Students use quantitative methods in programming problems and apply scientific-method reasoning by predicting program behavior, testing, interpreting results, and revising solutions. At least one graded programming task will use quantitative data or a model drawn from a scientific context.	Programming assignments; exams; final exam

4. Assignments and Learning Activities

- Programming Assignments / Homework: five individual graded assignments emphasizing design, implementation, testing, and clear documentation. Some assignments may contain smaller practice exercises or checkpoints.
- Short Quizzes: five short quizzes covering recent concepts, code reading, tracing, and basic problem solving.
- Midterm Exams: two exams assessing conceptual understanding, code tracing, program design, and short programming/problem-solving tasks.
- Final Exam: a comprehensive final examination during the official WCU final examination block.
- In-Class Practice: guided coding, tracing, debugging, and problem-solving activities are used throughout the semester. These activities also contribute to attendance/participation.

General Education artifact: Programming Assignment 5 (or an equivalent integrated assignment announced in D2L) will serve as the course's designated cumulative programming artifact. Retain a clean copy of your final submission and feedback.

5. Course Evaluation Policy and Grading

Evaluation	Weight	Planned Number
Attendance / Participation	5%	Ongoing
Short Quizzes	15%	5
Programming Assignments / Homework	30%	5
Midterm Exams	30%	2 (15% each)
Final Exam	20%	1

Letter Grade Scale

A = 90-100 | B = 80-89 | C = 70-79 | D = 60-69 | F = 0-59

Rounding: Final course averages are calculated from the weights above. Any rounding policy or extra-credit opportunity, if offered, will be stated uniformly for the class in D2L.

Nature and Timing of Feedback

- Programming-assignment feedback will normally be returned before the next major assignment of the same type is due.
- Exam 1 is scheduled early enough that students will receive substantive performance feedback before the Fall 2026 course-withdrawal deadline.
- Grades and feedback will be posted in D2L. Students should review posted grades promptly and contact the instructor about apparent errors.
- Course evaluations and final-exam materials will be retained in accordance with WCU requirements.

6. Programming, Collaboration, and Generative AI Guidelines

Programming Assignment Standards

- Submit your best version through D2L by the posted deadline. Code should be readable, appropriately indented, and documented according to the standards introduced in class.
- Partially working programs may receive partial credit when the submitted work clearly documents what works, what does not, and what you attempted. No submission receives no credit.
- Unless an assignment explicitly states otherwise, graded programming work is individual. You may discuss concepts and help another student identify an error, but you may not exchange solution code, jointly compose a solution, or submit another person’s work.
- You are expected to understand and be able to explain every part of the work you submit. The instructor may ask you to explain or modify submitted code as part of evaluating authorship and understanding.

Generative AI Tools

Generative AI can be useful as a programming tutor, but it can also bypass the practice needed to learn introductory programming. The following default policy applies unless a specific assignment states otherwise:

- Allowed: asking for conceptual explanations, interpreting an error message, requesting additional practice problems, or discussing general debugging strategies.
- Not allowed on graded programming assignments: asking an AI system to generate substantial portions of the solution, then submitting that output as your own work, unless the assignment explicitly authorizes such use.
- If AI-generated assistance is explicitly permitted for an assignment, disclose the tool and how it was used when requested, and be prepared to explain and modify the resulting code independently.
- Quizzes and exams: AI tools are not permitted unless the instructor explicitly authorizes them for that evaluation.

7. Course Policies

Attendance and Participation

Regular, on-time, prepared attendance is strongly expected because programming skills develop through repeated practice and immediate feedback. Attendance/participation is 5% of the course grade. University-excused absences will be handled consistently with WCU policy; students remain responsible for completing required academic work and for communicating about missed work as early as practical.

Late Work

Late assignments will be accepted without penalty when a valid excuse is communicated to the instructor before the deadline. Otherwise, unexcused late submissions receive no credit unless the assignment instructions explicitly provide a different late-work rule. The timestamp in D2L is authoritative for electronic submissions.

Email and Course Communication

Official course communication will use WCU email and D2L. Students are responsible for maintaining access to their WCU email account and checking D2L announcements. When requesting help, include enough information to identify the assignment/problem and describe what you have already tried.

Course Mode and Pre-Course Work

CSC 141-01 is scheduled as an in-person course. No graded coursework is assigned before the first official day of the semester.

8. Tentative Course Schedule

The schedule may be adjusted to match class progress. D2L announcements are authoritative for assignment due dates and minor sequencing changes. No coursework will be due beyond the university's permitted term dates.

Week / Dates	Topics	Planned Evaluation / Activity	Calendar Notes
1 Aug 25, 27	Course orientation; computational thinking; Python environment; expressions; print()	Practice: first Python programs	First class Tue Aug 25
2 Sep 1, 3	Variables, numeric and string types, assignment, console input/output	Programming Assignment 1 assigned	
3 Sep 8, 10	Boolean expressions; if / elif / else; program design	Quiz 1; Assignment 1 due	
4 Sep 15, 17	Nested decisions; testing; debugging; tracing	Programming Assignment 2 assigned	
5 Sep 22, 24	Functions; parameters; return values; decomposition	Quiz 2	
6 Sep 29, Oct 1	Function design; scope; systematic testing and debugging	Assignment 2 due; Exam 1 review	
7 Oct 6, 8	Iteration: while, for, range; accumulator patterns	Midterm Exam 1 (planned Oct 8)	Major feedback before withdrawal deadline
8 Oct 13, 15	Loops continued; nested iteration; integrated practice	Programming Assignment 3 assigned	Oct 13: Fall Break - no class
9 Oct 20, 22	Strings; indexing; slicing; traversal; sequence patterns	Quiz 3; Assignment 3 due	Oct 22: course-withdrawal deadline
10 Oct 27, 29	Lists and sequence processing; search/count/min/max patterns	Programming Assignment 4 assigned	
11 Nov 3, 5	Nested loops; data processing; introduction to file I/O	Quiz 4; Assignment 4 due	
12 Nov 10, 12	File input/output; text processing; command-line/console interfaces	Midterm Exam 2 (planned Nov 12)	
13 Nov 17, 19	Modular program design; integrated string/list/file problems	Quiz 5; Programming Assignment 5 assigned	
14 Nov 24	Integrated problem solving; quantitative/science-oriented programming application	Assignment 5 work / support	Nov 26: Thanksgiving Break - no class
15 Dec 1, 3	Cumulative review; final practice; course wrap-up	Assignment 5 due; final review	Last regular class meeting Thu Dec 3
Finals Dec 8	Comprehensive Final Exam	Final Exam, 10:30 AM - 12:30 PM	Official WCU block for T/R 11:00 AM course

9. Academic Integrity and University Policies

Academic Integrity

Students are expected to follow WCU academic-integrity requirements and the Computer Science Department's standards for academic honesty. Cheating includes plagiarism, submitting work prepared by another person, unauthorized collaboration or materials, copying during an examination, or otherwise misrepresenting authorship or performance.

- Department course rule used for this class: a first academic-dishonesty offense on an assignment may receive a zero on that assignment; a second assignment offense may result in an F for the course. Academic dishonesty on a test/exam may result in an F for the course, subject to applicable university procedures.

WCU Academic Integrity policy: Undergraduate Catalog - Academic Integrity

Office of Educational Accessibility (OEA)

West Chester University is committed to equitable access for students with disabilities. Students seeking accommodations should contact the Office of Educational Accessibility (OEA). Approved accommodations apply prospectively after the instructor receives the accommodation letter; they are not retroactive. Please share your approved accommodation letter as early as practical so arrangements can be made.

OEA: Wayne Hall 107 | 610-436-2564 | oea@wcupa.edu | OEA website

University-Sanctioned and Other Excused Absences

Notify the instructor in advance when possible about university-sanctioned events, required military service, or other absences covered by WCU policy. Appropriate documentation may be requested. The instructor will provide a fair opportunity to complete required work consistent with the applicable university policy.

Reporting Sexual Misconduct / Title IX

WCU faculty have reporting responsibilities for disclosures of sexual misconduct as defined by university policy. Students who need confidential support or want current reporting/resource information should consult WCU's Title IX and sexual-misconduct resources.

Emergency Preparedness

Students are encouraged to enroll in WCU ALERT for official emergency notifications. In an emergency, follow university instructions and classroom/building procedures.

WCU ALERT

Final Examination Policy

The course will use the official WCU final examination block. Individual instructors may not move a final examination outside the published time. For CSC 141-01 (T/R 11:00 AM), the published Fall 2026 block is Tuesday, December 8, 2026, 10:30 AM-12:30 PM.

APSCUF

I am a member of APSCUF, the Association of Pennsylvania State College and University Faculties. APSCUF supports excellence in teaching, scholarship, and service.

10. Classroom Learning Environment

Recording Policy

All classroom work, including (but not limited to) lectures, discussions, and/or group activities, is proprietary and not subject to public disclosure or distribution in the absence of express permission. Unauthorized recording of any part of this class (whether audio, video, or both) is strictly prohibited. Approval to record classroom activities may be obtained from the Office of Educational Accessibility (OEA) upon determination that recording is an approved accommodation. No recordings may be published, rebroadcast, or released in any form without my express permission. Violations may be addressed through applicable university procedures.

Academic Freedom

Consistent with the collective bargaining agreement and principles of academic freedom, faculty are entitled to freedom to teach, discuss, research, and publish in their academic field. Classroom discussion may include relevant material that is challenging or controversial when it advances the subject matter, debate, and learning.

Respectful Ethical Communication and Anti-Doxxing

To maintain a safe and ethical learning environment, students may not share or post personal information about the instructor or classmates, such as home addresses or personal contact information, without express permission. Classroom conduct and use of course materials must comply with relevant university policies.

Critical Thinking and Academic Inquiry

This course is designed to support critical thinking: examining evidence, considering relevant scholarly perspectives, asking questions, testing ideas, and forming your own conclusions. Course materials are selected for their relevance to the learning

outcomes. You are encouraged to question ideas, compare approaches, and develop reasoned positions. Grades are based on the stated assignment criteria, quality of reasoning, evidence, and technical correctness - not on whether your conclusions match those of the instructor or classmates.

11. How to Succeed in CSC 141

- Program frequently. Reading code is useful, but programming skill is built by writing and debugging code yourself.
- Start assignments early. Small misunderstandings compound quickly in programming, and early starts leave time to ask productive questions.
- When stuck, reduce the problem: reproduce the error with the smallest example you can, inspect variable values, trace control flow, and test one hypothesis at a time.
- Keep examples from class and your own corrected solutions. They are valuable study material for quizzes and exams.
- Ask for help before a deadline. Bring your current code, the exact error/output, what you expected, and what you have already tried.

This syllabus is a course plan. Reasonable adjustments may be announced in class and through D2L when needed to support course learning outcomes and the university calendar.