CSC 240: Computer Science III

Fall 2026 • Section 01 • CRN 34660

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, 9:30 AM - 10:45 AM
Location	Anderson Hall 315
Term Dates	August 24 - December 12, 2026
Course Site	Course website: wcu.ninja/csc240 (slides, schedule, materials). D2L: submissions, grades, announcements.

Syllabus version: August 24, 2026

1. Catalog Information and Course Description

Official Course: CSC 240 - Computer Science III (3 credits)

This course focuses on more advanced topics in object-oriented programming, including project design, planning, and testing using milestones and checklists. Programming topics include text processing (including StringBuilder and StringTokenizer classes), inheritance, polymorphism, abstract classes, interfaces, generic classes, exception classes, exception throwing and handling, random access files, serialization and an introduction to some basic data structures, such as collection classes and linked lists.

Prerequisite: Successful completion of CSC 142.

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

Catalog reference: WCU Computer Science course catalog

2. Textbook, Programming Language, and Tools

Textbook: Tony Gaddis, Starting Out with Java: From Control Structures through Data Structures, 4th Edition, Pearson, ISBN-13 978-0134787961. The 3rd Edition (ISBN-13 978-0134038179) is also acceptable; chapter/page references may differ slightly.

- Programming language: Java. Use the Java/JDK version specified in D2L for graded work.
- IDE: IntelliJ IDEA is recommended. Eclipse, NetBeans, jGRASP, or another instructor-approved Java IDE is acceptable.
- D2L: all assignments, announcements, supporting materials, and grades will be distributed or submitted through D2L unless otherwise announced. The public course website, wcu.ninja/csc240, posts the schedule, lecture slides, supporting materials, and the online CSC 141/142 diagnostic; graded submissions and grades remain in D2L.
- Regular access to a computer capable of compiling, running, debugging, and testing Java programs is required.

3. Course Student Learning Outcomes and Core Topics

Upon successful completion of CSC 240, students will be able to:

1. Plan, implement, test, and refine object-oriented Java projects using milestones, checklists, and systematic debugging. *[Program SLO 1, 2, 3]*
2. Process text using String, StringBuilder, StringTokenizer, wrapper classes, and related Java library facilities. *[Program SLO 1, 2]*
3. Design class hierarchies and correctly apply inheritance, method overriding, polymorphism, abstract classes, and interfaces. *[Program SLO 1, 2]*
4. Create and use generic methods and generic classes to improve type safety and code reuse. *[Program SLO 1, 2]*
5. Define, throw, catch, and propagate exceptions, including programmer-defined exception classes. *[Program SLO 1, 2]*
6. Use advanced file I/O techniques, including random-access files and object serialization where appropriate. *[Program SLO 1, 2]*
7. Explain and implement basic linked-list operations and use selected Java collection classes appropriately. *[Program SLO 1, 2]*

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.

Primary Content Sequence

- Chapter 9: Text Processing and More about Wrapper Classes
- Chapter 10: Inheritance, polymorphism, abstract classes, and interfaces
- Chapter 11: Exceptions and Advanced File I/O
- Chapter 18: Generics
- Chapter 20: Linked Lists and related collection concepts

4. Assignments and Learning Activities

- Programming Assignments / Homework: four individual Java programming assignments. Assignments emphasize design, implementation, testing, documentation, and the course topics listed above.
- Short Quizzes: two quizzes covering terminology, code reading/tracing, Java language features, and recent course concepts.
- Midterm Exam: one exam focused primarily on Chapters 9 and 10.
- Final Exam: a comprehensive examination covering the major topics of Chapters 9, 10, 11, 18, and 20.
- In-Class Practice: design exercises, code tracing, debugging, API/library practice, and short implementation activities are used throughout the semester and contribute to attendance/participation.

5. Course Evaluation Policy and Grading

Evaluation	Weight	Planned Number	Notes
Attendance / Participation	10%	Ongoing	On-time, prepared participation
Short Quizzes	10%	2	5% each
Programming Assignments / Homework	30%	4	HW1 6%, HW2 6%, HW3 12%, HW4 6%
Midterm Exam	20%	1	Primarily Chapters 9 and 10
Final Exam	30%	1	Comprehensive: Chapters 9, 10, 11, 18, 20

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.
- The midterm is scheduled early enough to provide substantive feedback before the latter portion of the semester.
- Grades and feedback will be posted in D2L. Students should review posted grades promptly and contact the instructor about apparent errors.

6. Programming, Collaboration, and Generative AI Guidelines

Programming Assignment Standards

- Submit your best version through D2L by the posted deadline. Code should compile when required, be readable and appropriately indented, and include documentation consistent with the standards discussed in class.
- Partially working programs may receive partial credit when the submitted work clearly indicates 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, trace, or modify submitted code as part of evaluating authorship and understanding.

Generative AI Tools

Generative AI can be useful as a programming assistant or tutor, but graded work must still demonstrate your own mastery of Java and object-oriented design. Unless a specific assignment states otherwise:

- Allowed: conceptual explanations, clarification of Java API behavior, interpretation of compiler/runtime errors, small illustrative examples unrelated to the graded solution, and general debugging strategies.
- Not allowed on graded programming assignments: asking an AI system to generate substantial portions of the required solution and submitting that output as your own work, unless the assignment explicitly authorizes such use.
- If AI-generated assistance is explicitly permitted, 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 explicitly authorized for that evaluation.

7. Course Policies

Attendance and Participation

Regular, on-time, prepared attendance is expected. Attendance/participation is 10% of the course grade. An unexcused late arrival or early departure of approximately 15 minutes may be treated as an unexcused absence. University-excused absences will be handled consistently with WCU policy; students remain responsible for completing required academic work and 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 programming help, include the assignment/problem, the relevant code or error message, the result you expected, and what you have already tried.

Course Mode and Pre-Course Work

CSC 240-01 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. The topic sequence follows the 2020 CSC 240 course structure while restoring the full in-person coverage of Chapter 11 and Chapter 20.

Week / Dates	Topics	Planned Evaluation / Activity	Calendar Notes
1 Aug 25, 27	Course orientation; review of Java/OOP foundations; project design, planning, milestones, checklists, and testing	Design/testing practice	First class Tue Aug 25

Week / Dates	Topics	Planned Evaluation / Activity	Calendar Notes
2 Sep 1, 3	Ch. 9: String processing; StringBuilder; wrapper classes	HW1 assigned	
3 Sep 8, 10	Ch. 9: StringTokenizer; parsing/tokenization; integrated text-processing problems	Quiz 1; HW1 work	
4 Sep 15, 17	Ch. 10: inheritance fundamentals; superclass/subclass relationships; protected members	HW1 due; HW2 assigned	
5 Sep 22, 24	Ch. 10: constructors, overriding, Object class, class hierarchies	HW2 work	
6 Sep 29, Oct 1	Ch. 10: polymorphism, dynamic binding, abstract classes, interfaces	Midterm review	
7 Oct 6, 8	Ch. 9-10 integration and review	Midterm Exam (planned Oct 8); HW2 due	
8 Oct 13, 15	Ch. 11: exception concepts; throw/throws; custom exception classes	Begin exceptions	Oct 13: Fall Break - no class
9 Oct 20, 22	Ch. 11: try/catch/finally; exception propagation and design	HW3 assigned	
10 Oct 27, 29	Ch. 11: advanced file I/O; random access files; serialization	HW3 work	
11 Nov 3, 5	Ch. 18: generic methods and generic classes; type parameters	Generics practice	
12 Nov 10, 12	Ch. 18: generics continued; type-safe reusable abstractions	Quiz 2; HW3 due	
13 Nov 17, 19	Ch. 20: linked-list concepts; nodes, references, traversal, insertion/removal	HW4 assigned	
14 Nov 24	Ch. 20: linked-list operations; selected collection-class connections	HW4 work / support	Nov 26: Thanksgiving Break - no class
15 Dec 1, 3	Integrated OOP design/testing; cumulative review; course wrap-up	HW4 due; final review	Last regular class Thu Dec 3
Finals Dec 10	Comprehensive Final Exam	Final Exam, 8:00-10:00 AM	Official WCU block for T/R 9:30 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. For CSC 240-01 (T/R 9:30 AM), the published Fall 2026 block is Thursday, December 10, 2026, 8:00-10:00 AM.

Registrar reference: Fall 2026 Final Examination Schedule

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 supports critical thinking through evidence-based program design, testing, debugging, comparison of implementation choices, and reasoned technical decisions. You are encouraged to question designs, compare approaches, and justify conclusions. Grades are based on stated criteria, technical correctness, quality of reasoning, and demonstrated understanding.

11. How to Succeed in CSC 240

- Do not treat inheritance, polymorphism, exceptions, generics, and linked structures as isolated syntax. Draw object relationships and trace runtime behavior before coding.
- Start programming assignments early. Advanced object-oriented bugs often span multiple classes and take time to localize.
- Use the debugger and small tests. Inspect object state, dynamic type, call paths, exceptions, and file state rather than guessing.
- Read Java API documentation directly and learn to distinguish library behavior from assumptions about your own code.

- Ask for help before a deadline. Bring the smallest reproducible example, 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.