CSCI 3160-001 — Computer Systems Syllabus

CRN: 83442 · **Credits:** 4 · Fall 2026

Instructor and Contact Information

Instructor: Chandler Scott

Email: scottcd1@etsu.edu

Office: Brinkley Center 135B

Drop-in hours: MTWR 11:00 a.m.–12:30 p.m., or by appointment.

Course Description

Introduces the abstractions that make C programs run on real machines: data representation, assembly, memory hierarchy, linking, processes, virtual memory, I/O, networking, and concurrency. Labs connect those topics to debugging and systems programming work.

Prerequisites

CSCI 2150 and CSCI 1260, with a grade of B- or better in CSCI 1260. Students also need working C and Unix programming experience sufficient to begin the course labs; the setup guide documents the supported environment.

Learning Outcomes

By the end of the course, students will be able to:

1. Interpret integer, floating-point, data-layout, and instruction representations and trace their effects on C program behavior.
2. Explain and diagnose how compilation, linking, loading, hardware, and operating-system services transform and execute programs.
3. Measure, analyze, and improve program performance using generated code, locality models, profiling evidence, and hardware constraints.
4. Trace and evaluate cache and virtual-memory behavior, including translation, protection, mappings, and memory errors.
5. Implement and debug programs using POSIX processes, signals, descriptors, sockets, threads, and synchronization, and trace pipe-based inter-process communication.
6. Design and justify correct systems software in C, including ownership, error handling, concurrency invariants, and security implications.

Required Materials and Technical Requirements

- **Required:** Bryant and O'Hallaron, *Computer Systems: A Programmer's Perspective*, 3rd ed., Pearson, ISBN `9780134092669`.
- **Supplemental:** Patterson and Hennessy, *Computer Organization and Design RISC-V Edition*, 2nd ed., ISBN `9780128203316`.
- Reliable internet access and a computer capable of running the course software.
- An x86-64 Linux development environment with a C compiler, debugger, and 32-bit compilation support. The Environment Setup guide documents supported local and emulated options.

D2L is authoritative for public handouts and the schedule. D2L carries announcements, private-repository clone URLs, and grades. Submit labs by pushing to the repository's `main` branch, not by

uploading code to D2L.

Course Structure and Major Assessments

Topics are introduced with examples, practiced through ICEs, and applied in systems programming labs. Expect to read C code, inspect program state with debugging tools, and explain behavior from evidence.

Assessment category	Weight
In-class exercises	10%
Programming labs	50%
Midterm exam	15%
Final exam	25%

In-Class Exercises

The nine graded ICEs practice each major CSAPP chapter or topic block before it appears in labs or exams. ICEs normally use a 45-minute block in the scheduled meeting; remaining time is reserved for lab work, debugging, or checkoff work. Environment setup is an ungraded preflight.

ICEs are graded on the reasoning process. Full credit requires an initial prediction or trace, enough intermediate state to make the reasoning visible, and a correction or evidence-based confirmation during comparison or debrief. Students normally work in pairs or groups of three and submit one paper with every member named before leaving the scheduled meeting. Contact the instructor for an approved accessible format or exceptional makeup route.

The highest eight of the nine ICE scores determine the ICE category grade. The one-score drop is the routine allowance for an absence or unusually difficult meeting; approved accommodations and documented exceptional circumstances may require a separate makeup route.

Programming Labs

Labs are the main programming work. Each student receives a provisioned private repository and its direct clone URL through D2L. Unless an approved extension says otherwise, each lab is due at 11:59 p.m. Eastern Time on its published due date. The repository state recorded at that cutoff is the submitted version.

The technical implementation contributes 80% of each lab grade, and the attached lab report contributes 20% using the common lab-report rubric. The six lab grades are weighted equally within the programming-lab category. Every lab repository must include a one-page `lab-report.pdf` answering the prompts in that lab's assignment.

Submitted code and written work must be individual. Students may discuss concepts, debugging strategies, and error messages, but may not view, copy, or share code, answers, payloads, or other assignment solutions. AI tools may explain concepts or diagnostics but may not generate or rewrite submitted solutions. Disclose permitted AI use as directed by the lab.

Exams

The midterm evaluates conceptual understanding and applied problem solving from the first half of the semester. The cumulative final covers system-level reasoning across the course; about one fifth of its points revisit pre-midterm foundations. The final exam is Wednesday, December 9, 2026, from 10:30 a.m.– 12:30 p.m.; the room is TBA and will be posted in D2L by December 2.

Grading Scale

Grade	Minimum percentage
A	93%
A-	90%
B+	87%
B	83%
B-	80%
C+	77%
C	73%
C-	70%
D+	67%
D	60%
F	below 60%

Calculated grades are not rounded by default.

Assignment and Assessment Policies

Lab pushes after the 11:59 p.m. Eastern cutoff are not graded unless an extension was approved. Request extensions before the deadline when possible; for emergencies, contact the instructor as soon as reasonably possible. Approved accommodations and university closures supersede the ordinary cutoff. Regrade requests should identify the specific issue and be submitted within seven calendar days.

Attendance and Participation

Attendance is not a separate grade category. ICE credit normally requires in-class participation and submission; the one-score drop provides routine absence flexibility. Treat classmates professionally in class, online, and during collaboration. Use clear email subject lines with the course number and section and include complete context for questions.

Academic Integrity and Generative AI

Students must follow the ETSU Honor Code. Academic misconduct is subject to university policy. Programming labs use the limited-assistance rule above: AI may explain concepts or diagnostics but may not generate or rewrite submitted solutions. Other assignments may define a different allowance. Disclose permitted AI use, including the tool and relevant prompts or interactions, and be prepared to explain all submitted work. Using another person's ideas, words, or work without attribution is prohibited, including internet sources and AI-generated text.

Communication and Course Technology

Use D2L for the schedule, public handouts, and announcements, private repository URLs, and grades, and Git for lab submissions. The course setup guide documents the required toolchain.

Accessibility and Student Support

ETSU accommodates students with documented disabilities. Contact ETSU Disability Services for approved accommodations. The official ETSU syllabus attachment provides university-wide information about accommodations, student rights, academic integrity, and support resources. Additional course

resources include BucsCARE, Sherrod Library, ITS technical help, D2L support, Microsoft Office 365, and Turnitin.

Course Schedule

The course schedule lists the complete meeting, lecture, ICE, lab, exam, and due-date sequence. The schedule and D2L announcements control schedule updates.

Syllabus Changes

Syllabus changes will be announced through D2L and reflected in D2L.

University Syllabus Attachment

Review the official ETSU syllabus attachment for university-wide policies, student rights, accommodations, academic integrity, and support resources in the official ETSU syllabus attachment.