CS333: Storage Systems

Syllabus for Fall 2026

General Info

Instructor: Bill Jannen
Email: 09wkj@williams.edu

Course Web Page: `http://www.cs.williams.edu/~jannen/teaching/f26/cs333`

Texts

We will be using the following textbook in this course:

- *Operating Systems: Three Easy Pieces* by Remzi H. Arpaci-Dusseau and Andrea C. Arpaci-Dusseau.

This book is available for free online, and PDF links to all readings are provided on the course webpage. If you prefer reading from a physical book, the campus bookstore has new and used copies, and the book's webpage includes links for ordering low-cost versions.

Since CSCI 237 is a prerequisite for this course, course materials assume that you are proficient in writing and debugging C code. Therefore, a good C reference is highly recommended. Copies of the following textbook are typically on reserve at the Schow Science Library:

- *The C Programming Language, 2nd edition.* by Brian Kernighan and Dennis Ritchie.

Additional readings will be assigned from various sources, including conference proceedings, research journals, magazines, and textbook excerpts. All required readings will be digitally accessible from the course website, although some may only be accessible while connected to the Williams network. If you are off-campus, you can access these readings using the college's VPN.

Course Description & Objectives

This course will examine topics in the design, implementation, and evaluation of storage systems. Topics include the memory hierarchy; ways that data is organized (both logically and physically); hardware characteristics and the ways that storage hardware influences storage software design; data structures and performance models; and system measurement/evaluation. The course emphasis will be on identifying and evaluating design trade-offs.

Upon the completion of this course, students will:

- be able to describe the performance profiles and behaviors of current and emerging storage technologies;
- when presented with a new storage technology, be able to describe how that technology will fit into or modify the "storage landscape";
- be able to critically read storage research texts and understand the texts' content and arguments;
- be able to describe and/or use common storage interfaces at the application, OS, and device layers;
- be able to read, write, document, debug, and test system code in C; and
- be able to analyze the performance of external memory algorithms and data structures.

Course Structure

Course meetings. This course will meet twice each week for lecture and discussion. *Attendance at course meetings is mandatory*. There are valid reasons to miss a course meeting, but *unexcused* absences will significantly reduce the participation component of your final course grade; additionally, missed in-class assignments will not be accepted.

Readings. There will be readings associated with each class, and you should complete all readings before their associated meeting. However, unexpected things happen during the semester. If you are unable to complete the readings, you should still attend the meeting and participate in discussions when able.

Labs. "Lab assignments" will be an important part of this course. Lab work is a way for us to explore select concepts at a deeper level. When completing labs, you should focus on completeness, correctness, and readability. Throughout the course, we will build practices that will help us evaluate our work on these axes, with a focus on modular design and incremental testing.

Collaboration. Collaboration is important, and this course will permit a variety of collaboration styles. Each assignment specification will explain the types of collaboration that are permitted for that assignment.

Midterm Exam. There will be one midterm exam. It will have two components:

- a 24-hour open-book component, and
- a closed-book component that will be in-class and timed.

Final Project. At the end of the course, you will either design your own project or select/customize a project proposal from a sample set that I will provide. In the first phase of your project, you will be required to submit a formal proposal that includes success criteria, methods for evaluation, and project deliverables. We will agree upon this proposal together. You will then complete your project, and submit your final write-up/artifact in place of a final exam.

Grading. Grades for this course will be determined as follows:

Preparation and participation:	15%
Lab assignments:	35%
Midterm Exam:	25%
Final Project:	25%

One goal of this course is to get you to think critically about what it means to write high-quality systems software. We should be able to:

- design and write tests that verify that our code is complete (i.e., it satisfies all stated requirements);
- write documentation that makes our code readable and maintainable (i.e., interfaces are well described with clearly chosen, meaningful parameters);
- describe a system's performance in concrete terms (i.e., expected run-times based on appropriate models for computation and I/O); and
- describe our system's design/architecture in a way that others can reason about without manually inspecting your code.

Honor Code

Collaboration is strongly encouraged, but it must be done in a way that promotes learning. To that end, each assignment will contain a description of the types of collaboration that are allowed, including any acceptable uses of generative AI.

Violations of course collaboration rules will be considered a violation of the college's honor code. If in doubt of what is appropriate, do not hesitate to ask your instructor.

Workload.

At Williams, we operate under the course unit system (rather than the credit hour system). You should expect to spend (on average) at least 13 hours per week on the academic and creative work related to class. This includes time spent meeting as a class and working on assignments. The Office of the Registrar explains the relationship of course units to credit hours in greater detail.

If you find yourself spending significantly more or significantly less time on this course, please let me know.

Course Calendar and Schedule.

The course calendar and schedule can be found on the course webpage, including links to the readings that should be completed prior to each course meeting (all required textbook materials are available as free PDF documents). The course webpage will be regularly updated after each course meeting to make all lecture materials and examples available for review.

Help!!!

Help. We all need it. There are many resources available when *you* need it. You are encouraged to discuss any questions, concerns, difficulties, or thoughts about the course with your instructor. If you find yourself facing challenges beyond the typical, I encourage you to reach out. Talk to your instructor, a friendly face from the Dean's Office, or some of the many professionals across campus who stand ready to help, including:

- The Peer Academic Support Network: For details, go to:
  `https://hub.williams.edu/peer-academic-support/content-tutoring/`
- Math & Science Resource Center: Support is available for students grappling with the more quantitative aspects of their coursework. For details, go to:
  `https://hub.williams.edu/peer-academic-support/quantitative-skills-programs/`
- Accessible Education and Disability Support Center: Some students with documented disabilities may require accommodations in certain situations. If that's you, take advantage of the options available. Go to
  `https://academic-resources.williams.edu/disabilities/` for details.
- The Health Center: Sometimes your challenges are not course-related. The Health Center provides a range of medical, psychological, and health/wellness services. Go to `https://health.williams.edu` for details.