

Logic for Systems

CSCI 195Y

Instructor: Tim Nelson
E-mail: tbn@cs.brown.edu

Mathematical logic provides the foundation for a rich set of tools for reasoning about systems and discovering whether their behavior meets our expectations. These tools allow us to model (e.g.) the state of buffers and caches, prove whether our protocols obey desirable properties, explore the consequences of memory-management strategies, and much more.

As a Computer Science student, you've often been asked to write code with the intent of *creating* a system. This class is different. Here, we'll ask you to create *models* of systems and interact with them in numerous ways. You will learn to use modern, logic-based tools to describe and analyze program designs, algorithms, data-structures, and other artifacts—we'll learn the logical frameworks we need as we go along. In the end, you'll develop a better understanding of how to use logic-based tools to analyze whatever systems you encounter after Brown.

Prerequisites: CSCI 016, 018, or 019. CSCI 022 (possibly concurrent) or instructor permission.

Required Text: We will assign readings that are either publically available or available to students from Brown's library online.

Time Requirements: Our class meets for 150 minutes per week and labs will consume up to 2 hours per week. You can expect roughly an additional 10–12 hours per week on assignments and projects.

Assignments and Grading

There will be no exams. Your course grade will depend purely on projects, the final project presentation, labs, and other assignments. The general breakdown is:

- 35%: Exercises (of varying weights) (formerly 30%)
- 15%: Midterm 1 (formerly 2 midterms for 30% total)
- 10%: SAT Assignment (formerly midterm 2)
- 25%: Final project
- 10%: Labs (of equal weight)
- 5%: Participation and professionalism

Because we update the course material every year, we sometimes find that assignments are harder (or easier!) than we expected. For this reason, *Tim only fixes letter-grade cutoffs at the end of the semester*. Before submitting final grades, Tim takes each student's overall performance trajectory into account.

Exercises and labs will be graded on a {check-plus, check, check-minus, zero} scale. A check means that you did fine on that assignment and are on track for an "A". A check-plus denotes exceptional performance. A check-minus means that there's significant room for improvement. Zeroes are given only for assignments that are missing or do not meet minimum functionality requirements.

Exercises Exercises will come in three flavors: assignments using tools we explore in class, programming problems, and written reflection assignments. Each exercise will be due at 11:59pm on Thursday nights and go out after class the preceding Friday. Students will receive *three* late days for use on exercises, each of which grants a 24-hour extension. Further late assignments will receive no credit.

Midterm and Final Project The course has two midterm projects and one final project. Students are required to form project teams well before the deadlines. Design checks with the course staff will occur at intervals before each due date. Each group will present their final project by appointment during the class presentation week in early May.

Note that you may not repeat project partners.

Extensions Extensions will be granted on a case-by-case basis in the event of illness (with a note from health services) or emergency. Please note that only the instructor can grant an extension.

More details on each assignment, project, etc. will be provided closer to the appropriate dates.

Policies and Schedule See the course webpage: <http://cs.brown.edu/courses/cs195y/>