

Logic for Systems
CSCI 1710 (*formerly 1950-Y*)
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 (concurrent is ok!) or instructor permission.

Required Text: We will assign readings that are either publicly available or accessible 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.

Grading

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 grade cutoffs at the end of the semester*. Before submitting final grades, Tim takes each student’s overall performance trajectory into account.

All assignments will be graded on a {check-plus, check, check-minus, zero} scale. Assignments with multiple components will receive a separate grade for each component. A check means that you did fine on that piece and are on track for an “A” (but see the section below on challenge problems). A check-plus denotes exceptional performance, and is rarely awarded. A check-minus means that there’s significant room for improvement. Zeros are given only for assignments that are missing or do not meet minimum functionality requirements.

Assignments Homework assignments will come in three flavors: exercises using tools we explore in class, programming problems, and written reflections. Each assignment will be due at 11:59pm Eastern time on Thursday evenings and go out after class the preceding Friday. Students will receive *five* late days for use on assignments, each of which grants a 24-hour extension. No more than two late days may be used on a single assignment.

Due to scheduling constraints related to grade-submission deadlines in the shortened 2021 Spring semester, **late days cannot be used on the final project.**

Challenge Problems

Most exercises will contain a “challenge problem”. In order to receive a grade of “A”, you must *make significant progress* on all challenge problems. “Significant progress” means attempting the problem and demonstrating a level of mastery, even if you don’t fully complete the problem. Challenge problems will receive a separate “check” designation from the main exercise.

Challenge problems may be submitted for checkoff up to three days after the corresponding assignment’s deadline. A late day used on an assignment will also grant an extra day for the challenge problem.

If your ability to submit a challenge problem on time is impacted by illness or other emergency, please see the Extensions policy below; our intention is to encourage students toward mastery, not lower grades on a technicality.

Even if you are taking the class S/NC, or do not want an “A” grade, it’s still worthwhile to do challenge problems: if you get the majority of challenge problems checked off, we will convert a check-minus to a check or a zero to a check-minus—effectively making challenges extra credit.

Final Project The course has one final project. Students are required to form project teams well before the deadline. 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.

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/cs171/>

Last Modified: Jan 13th 2021