Mathematical logic provides the foundation for a rich set of tools for reasoning about systems and discovering whether their behavior meets our expectations. In spite of how it is often presented, logic is not only good for describing the state of Socrates and rain, but also the state of buffers and caches and the other difficulties that everyday practicing computer scientists wrestle with.

In this course, you will learn to use modern, logic-based tools to describe and analyze program designs, algorithms, data-structures, and other artifacts—learning the logical frameworks we need as we go along.

**Prerequisite:** CSCI 016, CSCI 018, or CSCI 019.

**Preferred but not required:** CSCI 022 and CSCI 1010 (formerly 051).

**Required Text:** Software Abstractions (2nd edition) by Daniel Jackson (MIT Press)

**Time Requirements:** Our class meets for three hours per week, and we have four 90-minute labs spaced throughout the semester. You can expect an additional 8 to 10 hours per week on assignments and projects, with some requiring more time than others.

**Assignments and Grading**

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

- 60%: 9 Assignments (roughly 6.67% each)
- 8%: 4 Required Labs (2% each)
- 6%: Case-study review
- 26%: Course project

This class does not grade on a curve. However, because we update the course material every year, we sometimes find that assignments are harder (or easier!) than we expected. For this reason, we fix letter-grade cutoffs only after examining a sample of student work on each assignment.

**Assignments** Assignments will come in two flavors: exercises that use tools we explore in class and programming assignments. Each assignment will be due at 11:59pm on Thursday nights and go out after class the preceding Friday. We will accept one late assignment, within 24 hours, for full credit. Further late assignments will receive no credit.

**Case-Study Reviews:** Our text contains several software modeling case-studies. Students will each pick a partner and then select two different case studies by **March 2nd**. You will present these, code-review style, on **March 20th–24th**. Be prepared to explain and constructively criticize the case-studies you pick.

**Course Project** Students each pick a different partner by **March 16th**. Project topics can be anything touching on modeling and verifying software or systems and should apply one of the tools we have used in class. Prior topics have included data-structures, protocols, user-interfaces, and more. Project proposals are due on **March 23rd**. There will be required design checks with the course staff prior to approval. Projects (source, along with a short 1 page writeup) are due electronically on **May 2nd**. Each group will then present their final project, in a format similar to the case-study reviews, by appointment during **May 3rd–5th**.

**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/2017/](http://cs.brown.edu/courses/cs195y/2017/)