# CS 270 Mathematical Foundations in CS - Syllabus

**Term and Credits**

Spring 2018-2019
3 Credits

**Room and Time**

- Section 1 - Monday and Thursday – 11:00am -1:50pm 3675 Market, Room 1054-1055
- Section 2 – Monday and Thursday – 3:00pm – 4:50pm 3675 Market, Room 1054-1055

**Instructors**

Professor Bruce Char (Section 1)
*Electronic Mail Address:* charbw@drexel.edu
*Office:* 3675 Market St. Room 1061
*Extension:* 215-895-2670
*Office Hours:* Mon and Thurs 2-3pm

Professor Mark Boady (Section 2)
*Electronic Mail Address:* mwb33@drexel.edu
*Office:* 3675 Market St. (new home of CCI), Room 1058
*Extension:* 215-895-2347
*Office Hours:* Tues 2-5pm, Thurs 1:30-2:30

**Teaching Assistant(s)**

- Steve Earth, se435@drexel.edu. Office hours: Tues 4-6pm, Fri 2-4pm
  Additionally, Steve can be available by appointment in his PhD lab in #1143
- Vivian Zhou, yz534@drexel.edu, Office hours: Tues noon-2pm, Wed noon-2pm
- Bao Pham, btp38@drexel.edu. Office hours: Wed noon-2pm

All TA Office Hours held in the CLC https://www.cs.drexel.edu/clc As of 4/1/2019 it will be located at 3675 Market St. Room 1066. The CLC staffing includes many other times other assistants who can answer CS270 questions will be on duty.


Steve Vivian Bao

**Course Description**

Introduces formal logic and its connections to Computer Science. Discusses propositional and predicate logic, logical inference, recursion and recursively defined sets, mathematical induction, and structural induction. Students learn to translate statements about the behavior of computer programs and computational situations into logical claims and to prove such assertions using both traditional techniques and automated tools. Considers approaches to proving termination, correctness, and safety for programs.

**Course Objectives and Goals**

1. To learn how to work in computer science contexts using symbolic/mathematical representations.
2. To learn how to use symbolic logic to describe the state of systems
3. To learn how to use logical deduction (by hand and using tools) to prove properties of systems.
4. To be able to use recursion and divide and conquer to solve problems.
5. To be able to provide recursive definitions of patterns and data structures.
6. To be able to formally specify the input/output requirements of programs.
7. To be able to use induction and other proof techniques to prove properties of algorithms, data structures, programs, and computer systems
8. To understand the power and limitations of formal logic.

**Audience and Purpose within Plan of Study**

This is a required course for all Computer Science and Software Engineering students. It should also be of interest to Computer Engineering, Mathematics students and students with an interest in logic and computation.

**Prerequisites**
Students should have programming experience (CS172 or equivalent). High school and college-level mathematical knowledge (arithmetic with whole numbers and fractions, what a function is, high school algebra, geometry), reasoning (justifying why an algebraic derivation or geometric proof works) and computational skill will be used in the course, so it's expected that you already have some experience and proficiency in this.

**Specific Computer Science prerequisites students should know prior to this course**

1. Ability to read and understand code.
2. Be able to write code to solve simple problems (at the level of cs 171-172).
3. Basic understanding of program execution, including definition and invocation of functions/procedures.
4. Ability to write simple recursive programs.
5. Be able to construct and evaluate Boolean expressions as used in if or while statements in conventional programming languages.

**What Students will be able to do upon successfully completing this course:**

1. To be able construct statements in symbolic logic describing situations in every day life or computer science contexts.
2. Use proofs by deduction to justify logical statements
3. Be better able to write and analyze recursive functions
4. Be able to implement and use a SAT solver.
5. Use Inductive Proofs to justify the correctness of programs and statements.
6. Use logic to describe the properties of systems.

**Textbook**

We will use free resources for this class, available through the BBLearn class website.

**Optional Resources**

Book of Proof (Second Edition)
Richard Hammack
Paperback: ISBN 978-0-9894721-0-4
Hardcover: ISBN 978-0-9894721-1-1
Available for Free online at: http://www.people.vcu.edu/~rhammack/BookOfProof/

The Racket Guide
Matthew Flatt, Robert Bruce Findler and PLT
https://docs.racket-lang.org/guide/index.html

# Course Material

### Programming Language

- This class will use the Racket Programming Language.
- This class will use the DrRacket IDE for development.
- Download Racket

### Lectures

- Lectures will be held twice a week.
- Attendance in lecture is required.
- If you need to miss class, please message the Professor to request an exception.

### Labs

- Almost all classes will have an in-class lab. Lecture period includes lab exercises which must be completed and turned in at the end of class in order to get credit. If you cannot complete the exercise by the end of class, you should hand in what you have completed. Unless there is an excused absence, missing a lab means not getting credit for it, with no chance for make up.
- You are free to think and work further on lab exercises after class is (encouraged because of the learning effect) but you will not get more credit for that out of class work unless an exception is granted by the professor. Since course grading will be "curved" how much you turn in matters mainly if you complete a lot less relative to others in class in the time allowed.
- Exercises will be require working with a team.
- If you need to miss class, please message the Professor to request an exception.

### Quizzes

- Quizzes will be given in class on the dates indicated in the calendar below.
- Quizzes are closed book.
- Quizzes will have a time limit, typically 15-20 minutes.

### Homeworks

- Most weeks will have a homework assignment.
- Homeworks are due Fridays at 11:59PM
- Homeworks must be submitted to learning.drexel.edu
- **Late Submissions:**

- 1 Day Late 10 percentage points deduction (Before Saturday 11:59PM). For example, if you got 69% but were a day late, you'd get a 59%.
- 2 Days Late 20 percentage points deduction (Before Sunday 11:59PM)
- 3 Days Late 30 percentage points deduction (Before Monday 11:59PM)
- 4 or More Days Late - Only Accepted with Instructor Permission

**Exams**

- There will be two exams for the class
- Midterm during Week 5 (Thursday, May 2nd).
- Final Exam during Exam Week (Week 11)

**Special Circumstances**

- If you have a documented reason why you cannot submit an assignment by the deadline, a special exception may be made. The Professor may also wave the late submission penalty for documented special exceptions.

# Course Policies

**Academic Honesty Policy**

The CCI Academic Honesty policy is in effect for this course. Please see the policy at http://drexel.edu/cci/resources/current-students/undergraduate/policies/cs-academic-integrity/.

Academic Honesty Violations will be reported to the University. Punishment will be determined by the severity of the incident. Punishments include, but are not limited to,

- Failing grade for class, without opportunity to withdraw.
- Deduction of one letter grade
- Zero on Assignment/Exam where violation occurred.

**Grading and Policies**

- Labs 19% (19 labs)
- Homeworks 20% (8 homeworks)
- Quizzes 16% (4 quizzes)
- Midterm 20%
- Final Exam 25%
- Pre-, post- course surveys. 0.5% extra credit each.

Final grades will be determined by your total points weighted according to this distribution. Grades may be curved. It may be modified at the instructor's sole discretion. If you are concerned about your grade, speak to the instructor.

**Computer/Software Help**
iCommons: http://drexel.edu/cci/about/our-facilities/rush-building/iCommons/

**University Policies**
In addition to the course policies listed on this syllabus, course assignments or course website, the following University policies are in effect:

- Academic Honesty: http://www.drexel.edu/provost/policies/academic_dishonesty.asp
- Judicial Affairs Academic Integrity: http://drexel.edu/studentlife/community_standards/facultystaff/integrity/
- Official Final Exam Schedule: http://www.drexel.edu/registrar/scheduling/exams/
- Students with Disability Statement: http://drexel.edu/oed/disabilityResources/overview/
- Course Drop Policy: http://www.drexel.edu/provost/policies/course_drop.asp
- Drexel Student Learning Priorities: http://www.drexel.edu/provost/irae/assessment/outcomes/dslp/
- The instructor may, at his/her/their discretion, change any part of the course during the term, including assignments, grade breakdowns, due-dates, and the schedule. Such changes will be communicated to students via the course web site Announcements page. This page should be checked regularly and frequently for such changes and announcements. Other announcements, although rare, may include class cancellations and other urgent announcements.

**Avoiding Academic Honesty problems**

Students are expected to do their assignments on their own except when permission is given to create solutions together, such as with group exercises. The course credit given for an assignment is in part for an authentic demonstration that you can be original and inventive – that you can do more than imitate, look up, and copy. What did you learn to do in an assignment that won't be done by a codebot next year (https://qz.com/920468/artificial-intelligence-created-by-microsoft-and-university-of-cambridge-is-learning- to-write-code-by-itself-not-steal-it/)?

We observe that some students believe (probably a holdover from bad high school math education, or taking too many multiple choice tests) that all correct answers look the same – that if 47 is the final result, that all the thinking that produced it is irrelevant or not considered in the assessment of the assignment. Yet college-level work is far from that – even programs or answers to math problems are expected to be written for easy comprehension, with justification and explanation given along the way. Otherwise even experts won't understand what you are talking about. At the college level, technical answers become more like essay writing than producing a single formula or expression. So "signature" moves, formatting or phrasing will be expected for most work that you'd turn in. If you have real understanding of a subject, you should be able to explain it again, "live", as well if not better as you did in writing.

When submitting an assignment, you have to be honest about where you got ideas, code, and information. Where we expect individual invention it would be wrong (and a violation of the

academic honesty policy) for you to claim that you invented something that you didn't. To avoid any troubles, all code or other responses should have proper attribution as to the source of the ideas. This includes but is not limited to: other books or papers, sources on the Internet, information you get from talking or sharing with other people, code that you clone or modify but didn't create from scratch. To properly attribute, you should include citations that make it easy for a reader to verify "this code or answer was derived from this source". Usually in courses there is a blanket exception about needing to attribute to code copied from the course materials themselves, but only because all the readers know about those sources and can easily determine what originated from there. But everything else – stuff from stack overflow, other textbooks or web pages, cloud file repositories, others you get to produce answers for you, etc. should be attributed or you risk penalties. **The prime directive is: if you take information from other sources, you must give proper attribution in what you submit.**

Students who do not meet this standard will be in violation of Drexel's Academic Honesty policy. You can read more about it at http://www.drexel.edu/provost/policies/academic_dishonesty.asp. Penalties that will be imposed in the case of cheating, plagiarism, or givings other assistance as to how to write submissions, etc. include: reduced marks (or zero) for the assignment or test, reduction in final grade, or failure in the course with no option to withdraw. The College of Computing and Informatics has a "two strikes and you're out" policy for its majors who violate the policy.

If you feel that you're relying too much on others to get the work done, talk to the instructor. Hopefully they will be able to suggest things that will allow you to be more self-sufficient. If you are not sure about how to satisfy the Academic Honesty expectations for some course work, talk to the instructor. It's your responsibility to do this before trouble starts.

The "prime directive" is**: if you use information beyond the approved course information sources** (the readings, lectures, the TAs and instructors, handouts, course readings**) you must list its source** (give proper attribution) in a clear way in what you hand in.

**Topics**

1. Functional Programming
2. Recursion, Recursive Definitions and Induction
3. Propositional and Predicate Logic
4. Formal Proof using Natural Deduction
5. Applications of Logic to Computer Science
6. Divide and Conquer Algorithms and Recurrence Relations
7. Program Specification and Verification
8. Automated Reasoning
9. Termination Analysis
10. Test Case and Counter Example Generation

# Tentative Course Schedule

Please see the appropriate assignment webpages for a detailed description of course deliverables.

| Week | Topic | Supplemental Links | Lab | Homework |
|---|---|---|---|---|
| 1 (4/1/19) | Introduction to Racket | Quick: An Introduction to Racket with Pictures<br>So You Want to be a Functional Programmer (Part 1) | Lab 1 and Lab 2 | Course pre-survey, Homework 1 - Due 4/5/19 at 11:59PM |
| 2 (4/8/19) | List Processing and Natural Numbers | List, Iteration, and Recursion<br>High-order list operations (Ignore Haskell Part)<br>Peano Axioms | Lab 3 and Lab 4 | Homework 2 - Due 4/12/19 at 11:59PM<br>Quiz 1 – In class 4/10/19 |
| 3 (4/15/19) | Boolean Expression and Prepositional Logic | Chapter 2.1-2.6 from Book of Proof | Lab 5 and Lab 6 | Homework 3 - Due 4/19/19 at 11:59PM |
| 4 (4/22/19) | Predicate Logic | Chapter 2.7 from Book of Proof<br>Predicate Logic | Lab 7 and Lab 8 | Homework 4 - Due 4/26/19 at 11:59PM<br>Quiz 2 – In class 4/25/19 |
| 5 (4/29/19) | Specification and Logic | | Lab 9 and Lab 10 | In-class midterm on 5/2/19 |
| 6 (5/6/19) | Midterm + SAT Solver | MiniSat in Browser<br>Boolean Satisfiability Problems | Lab 11 and Lab 12 | Homework 5 - Due 5/9/19 at 11:59PM |
| 7 (5/13/19) | Natural Deduction | Chapter 4 from Book of Proof<br>Deduction Proof Checker<br>Pages 142 to 164 of Symbolic Logic: A First Course | Lab 13 and Lab 14 | Homework 6 - Due 5/17/19 at 11:59PM<br>Quiz 3 – In class 5/15/19 |
| 8 (5/20/19) | Proofs by Contradiction | Chapter 6 from Book of Proof<br>Deduction Proof Checker | Lab 15 and Lab 16 | Homework 7 – 5/24/19 at 11:59PM |

| | | | | |
|---|---|---|---|---|
| | | Pages 164 to 183 of Symbolic Logic: A First Course | | |
| 9 (5/27/19) | Mathematical Induction<br><br>**No class 5/27: Memorial Day** | Chapter 10 from Book of Proof | Lab 17 | Homework 8 – Due 5/31/18 at 11:59PM |
| 10 (6/3/19) | Structural Induction | | Lab 18 and Lab 19 | Quiz 4 – In class 6/3/19<br>Course post-survey (online) |
| 11 (6/10/19) | **Final Exam**<br>Location t.b.a. (look it up on the University Final exam schedule when it is posted). | | | |