

CS103A Syllabus & Information

Instructor: Maggie Johnson
Email: johnson@cs
Office Phone: 723-9798
Office Hours: Monday, 2-5 in Gates 188

Please send all questions to cs103a@hotmail.com. Be sure to monitor the newsgroup frequently: su.class.cs103.

CS103A - Discrete Mathematics

CS103A is the first of a two-quarter sequence (CS103B is next quarter). Both courses together constitute an in-depth course in discrete mathematics. This is the area of mathematics that deals with the study of discrete objects, where “discrete” means distinct or unconnected. Discrete math is used, for example, whenever objects are counted, when relationships between finite sets are studied, and when processes involving a finite number of steps are analyzed. This area of math has become increasingly important because information is stored and manipulated in a computer in a discrete fashion.

Discrete math provides the mathematical foundations for many computer science courses including data structures and algorithms, compilers, automata theory and formal languages, operating systems, database theory, to name a few. You will find these courses much more difficult if you attempt them without the foundations of discrete math.

Our goal in this course is to build skills and give you experience in the following areas:

1. **Mathematical Reasoning:** The ability to construct a sound logical argument is essential for computer scientists, not only because proofs are important in certain areas of computer science, but also because the same basic thought process is used in constructing a proof and in writing a program.
2. **Combinatorial Analysis:** An important problem solving skill is the ability to count or enumerate objects. It pops up surprisingly often in computer science applications.
3. **Discrete Structures:** These are the abstract mathematical structures used to represent discrete objects and relationships between those objects. Discrete structures include sets, permutations, relations, trees, graphs and finite-state machines. These structures form the conceptual basis for many of the data structures that we use as programmers.
4. **Algorithmic Thinking:** Certain classes of problems are solved by the specification of an algorithm that can be implemented in a program. The mathematical portions of this activity (which will interest us most) include the specification of the algorithm, the verification that it works properly, and the analysis of the computer memory and time required to perform it.
5. **Applications and Modeling:** Discrete math has applications to almost every conceivable area of study including (of course) computer science, chemistry, botany, zoology, linguistics, geography, business, etc. Modeling with discrete math is an extremely important problem-solving skill.

In summary, our primary tasks in this course are to develop your proving, problem-solving and algorithmic skills, and to use discrete structures as abstract models for use in solving problems and developing algorithms.

Pre-Requisites

There is no pre-requisite for this course but the student is assumed to have some experience with programming.

Lectures

MWF, 10 a.m. in 420-040.

Sections

There will be a weekly section where you can meet with a TA for an hour in small groups. The primary purposes are to review material from the lectures and/or to work problems that exercise your understanding. Participation in sections is not required, but can be very helpful especially if the material is new to you. Section time and location will be announced the first week of the quarter.

“Working” Office Hours

Each week, one of our TA’s will hold office hours in a classroom where you can come and work on problems from a problem set or the textbook. The TA will be available to answer questions when needed. The purpose of this is to provide students with additional options for obtaining assistance besides section, and one-on-one office hours.

Problem Sets & Exams

There will be weekly problem sets that will not be graded. We will provide solutions to these problem sets so you may check your own work. *It is essential that you work through all the problems on the problem sets - you will not pass the exams without doing so.* We prefer that you do your own work on problem sets since you will be on your own when you take the exams.

The reason for this class structure is we want you to obtain feedback on your solutions as soon as possible after you have done the problem. It’s been our experience that this feedback is essential in building strong skills. If your solution differs from our solution, we are providing many opportunities (section, office hours, the help line) for you to contact us with your questions. It is your responsibility to stay on track. *It will simply not be possible to do all the problems on the problem sets right before the exam.* The skills that are developed in doing the problem sets are built up gradually. It cannot be done in one night.

There are four exams in CS103A. The first three are 24-hour take home exams, and the last is an in-class final. Exams must be taken during the 24-hour period specified in the syllabus, so please plan your time accordingly.

Grading

Final grades will be based on the following:

15%	Exam 1
25%	Exam 2
25%	Exam 3
35%	Final

Textbook

There are two required textbooks that you can purchase in the bookstore. Additional materials will be provided in class. If you miss picking these things up in class, you can obtain them from the handout bin in Gates (if any are left over), or from our web page.

There will be regular reading assignments and most of our problem sets will come from the textbooks. Lecture materials will sometimes cover material in the textbook, but will more often present supplementary materials and additional examples.

Rosen, K., *Discrete Math and Its Applications*, New York: McGraw-Hill, 1999, and the *Student Solutions Guide*.

Barwise, J. & Etchemendy, J., *Language, Proof and Logic*, New York: Seven Bridges Press, 1999.

How to Succeed

The best way to obtain the skills required to succeed in this course is to solve problems - lots and lots of problems. The textbooks provide many practice problems with detailed solutions given in the Solutions Guide (for Rosen). The Solutions Guide also has a guide to writing proofs, a list of common mistakes, sample exams, etc. So the way to succeed in CS103A: come to class where you will see especially important examples and applications, *stay on schedule* by doing a part of the current problem set each day after class, study the textbook and solutions guide carefully and thoroughly, and come and see us if you need help (or write to the help line). Most importantly, if you are feeling confused, frustrated or worried about anything pertaining to the course, let us know.

Honor Code

You may do problem sets anyway that works best for you. It's best if you work on the problems yourself since this is the best preparation for the exams. You may discuss problem set questions with a partner or a study group. Just remember, you have to be able to do the problems yourself.

Exams are to be done individually and must represent original work - it is a violation of the honor code to copy or derive exam question solutions from anyone, from textbooks, or from previous instances of this course.

CS103A Syllabus

Date	Topic	Due
9/27	Introduction	
9/29	History and Background	
10/2	Introduction to Prop Logic	Read BE pp. 1-10, 19-31, 41-46
10/4	Methods of Proof I	Read BE pp. 46-65

10/6	Connectives	Read BE pp. 67-86, 93-100
10/9	Logical Consequence	Read BE pp. 106-125
10/11	Boolean Algebra	
10/13	Simple Circuits	
10/16	Methods of Proof II	Read BE pp. 127-141
	Exam 1: distributed noon 10/16, due noon 10/17	
10/18	Proof Strategies	Read BE pp. 167-173
10/20	Conditionals	Read BE pp. 176-190
10/23	Quantification I	Read BE pp. 227-251
10/25	Logic of Quantifiers	Read BE pp. 257-283
10/27	Multiple Quantifiers	Read BE pp. 289-307
10/30	Proofs with Quantifiers I	Read BE pp. 319-338
11/1	Proofs with Quantifiers II	Read BE pp. 342-36
	Exam 2: distributed 11/2, due noon 11/3	
11/3	No Class	
11/6	Resolution	
11/8	Proving Real Theorems I	Read Rosen pp. 167-181
11/10	Proving Real Theorems II	
11/13	Sequences and Summations	Read Rosen pp. 69-76
11/15	Induction I	Read Rosen pp. 186-199
11/17	Induction II	

11/20	Program Proofs	
	Exam 3: distributed 11/20, due noon 11/21	
11/22	Recursive Definitions	Read Rosen pp. 202-209
11/27	Recursive Algorithms	Read Rosen pp. 214-218
11/29	Combinatorics I	
12/1	Combinatorics II	
12/4	Functions	Read Rosen pp. 56-67
12/6	Review	

Final: Dec. 15, 8:30 a.m.

I. Basic Tools

- 0. Background and History
- B. Logic and Proof
- C. Boolean Algebra
- D. Induction
- E. Combinatorics
- F. Recursion
- G. Functions
- H. Analysis of Algorithms
 - 1. Big-O
 - 2. Recurrence Relations
 - 3. Analysis of Non-Recursive & Recursive Programs

II. Discrete Structures

- A. Sets
 - 1. Concepts and Definitions
 - 2. Infinite Sets & Countability
- B. Relations
 - 1. Concepts and Definitions
 - 2. Relational Database Theory
- C. Linear Structures
 - 1. Concepts and Definitions
 - 2. Applications
- D. Trees
 - 1. Concepts and Definitions
 - 2. Structural Induction
 - 3. Applications
- E. Graphs
 - 1. Concepts and Definitions
 - 2. Paths and Circuits
 - 3. Shortest Path Algorithms
 - 4. Applications

III. Introduction to Theory

- A. Regular Expressions
- B. Finite Automata
- C. Context Free Grammars
- D. Pushdown Automata
- E. Turing Machines
- F. Undecidable Problems