Handout #1 | CS103A
September 22, 2008 | Robert Plummer

CS103A Course Information—Autumn 2008

Course Title: Discrete Mathematics for Computer Science

Units: 3

Lectures: MWF 1:15 – 2:05 P.M. in Gates B03. Lectures are also available through Stanford Online.

Instructor: Dr. Robert Plummer

Email:	plummer@cs.stanford.edu
Office:	Gates 178 (ground floor, B wing)
Office hours:	Mon., Wed. 2:15 –4:15, or by appointment
Office phone:	723-4350

TA: Brian Eggleston

Email:	briane@cs.stanford.edu
Office:	TBA
Office hours:	TBA
Office phone:	TBA

TA: Tyler Mullen

Email:	tmullen@stanford.edu
Office:	TBA
Office hours:	TBA
Office phone:	TBA

TA: Sudeep Tandon

Email:	sudeept@stanford.edu
Office:	TBA
Office hours:	TBA
Office phone:	TBA

TA: Tiyu Wang

Email:	tiyuwang@gmail.com
Office:	TBA
Office hours:	TBA
Office phone:	TBA

Website: **http://cs103a.stanford.edu**

The class website will be used for important course announcements, and it is a place where you can pick up any handouts that you have missed. It is your responsibility to check the website periodically.

Email hotline

To get fast answers to question about this course, send email to cs103a@gmail.com. This address is checked frequently by course staff and is your best source of help. Please note that when you begin submitting homework electronically, there will be a different email address for that. This one is just for asking questions.

Pre- and Co-Requisites

CS106A is a pre- or co-requisite of this course.

"Working" Office Hours

Each week, 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 the help line. The location and times will be announced.

Problem Sets & Exams

There will be several problem sets that will be graded. During the first half of the course, most of the problem sets will be submitted online and graded using an automated grading system. Later in the course, problem sets will be hand-written and hand-graded. Each student is to work individually on the problem sets. You are allowed to work together with certain restrictions; be sure you read the Honor Code section below. All problem sets are due at class time on the dates specified in the syllabus.

There are two exams in CS103A. The midterm is a 2-hour in-class exam and the final is a 3-hour in-class exam. The mid-term will be held **Thursday, Oct. 23, 7:00-9:00 pm**. The final will be held **Wednesday, Dec. 10, 8:30-11:30 am**. Locations will be announced. **If you take this course, you are obligated to take the exams at the published times.** In particular, students are sometimes tempted to take two courses that meet at the same time if one of them is on SCPD. This leads to examination conflicts, and this practice is not condoned by the CS Department. An alternate final will not be given to eliminate such conflicts.

Late Policy

Late assignments will be accepted only up to the class period following the due date. This allows us to publish solutions in a timely manner. The penalty for a late assignment is 10% of your grade for that assignment. However, each student is allowed 3 "free" late days; that is, you may turn in up to three assignments one class period late without penalty. After you use up your three free days (on three different assignments), the 10% penalty will apply. Extensions may be granted under extenuating circumstances but must be requested *prior* to the due date. All such requests must be made directly to the professor, not the TA's. **To repeat: with or without free late days, no assignment will be accepted more than one class day after the due date. You should also be aware that Problem Set 6 (just before the midterm) is a special case that is not accepted late.**

Grading

Final grades will be based on the following:

45%	Problem Sets
20%	Midterm
35%	Final

To receive a passing grade, you must complete passing work on both the problem sets and the exams.

Textbook

There is one required textbook, which you can purchase in the bookstore. Additional materials will be provided in class. If you miss picking these up in class, you can obtain them from the handout bin in Gates near Room 182 (if any are left over), or from our web page. The textbook is:

Barwise, J. & Etchemendy, J., *Language, Proof and Logic*, Stanford: CSLI Publications, 2007.

In the second half of the course, readings will be from handouts. If you would like a supplemental text, a good choice is the following (check Amazon.com):

Rosen, Kenneth H., *Discrete Mathematics and Its Applications, 6th Ed.*, McGraw Hill, 2007.

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. Come to class where you will see especially important examples and applications, and *stay on schedule* by doing a part of the current problem set each day after class. The problem sets are frequent in the first half of the quarter (due every 5 days). Use the textbook for additional information and extra problems as needed; 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, please let us know.

Honor Code

Problem sets are to be done "from scratch", i.e., it is a violation of the honor code to copy problem set or exam question solutions from anyone, including: other students, textbooks, the internet, or from materials from previous instances of this course. Discussion of problem sets is allowed, but you should keep these discussions at the strategic level as far as possible. Be aware that if you reveal the key idea of a problem to another student, you are denying that person the opportunity to have the "Aha!" experience, which is where the greatest learning takes place. If you receive such information, you are likely to do less well on exams than if you figure things out for yourself. Even if you do work with a study group, when it comes time to write up your solution, you must do so by yourself.

We should also mention that the automated grading system we use for the first half of the quarter automatically checks if your work matches not only other students in this course, but students who have previously taken the course. The instructor and TA's are notified when such a match occurs.

If you obtain substantial help from a TA or another student, you must document this on your problem set. Finally, a good guideline is you must be able to explain and/or duplicate anything that you submit.

Course Description

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 reasoning skills are used in both constructing proofs and in writing programs.
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.

Outline of topics for CS103A/B

I. Basic Tools

1. Formal Logic and Proof Techniques
2. Number Theory and its Applications
3. Proving "Real" Theorems
4. Induction
5. Program Proofs
6. Recursion
7. Combinatorics & Probability
8. Functions

↑ CS103A

↓ CS103B

9. Analysis of Algorithms
   a. Big-O
   b. Recurrence Relations
   c. 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. Spanning Trees
  - 5. Applications

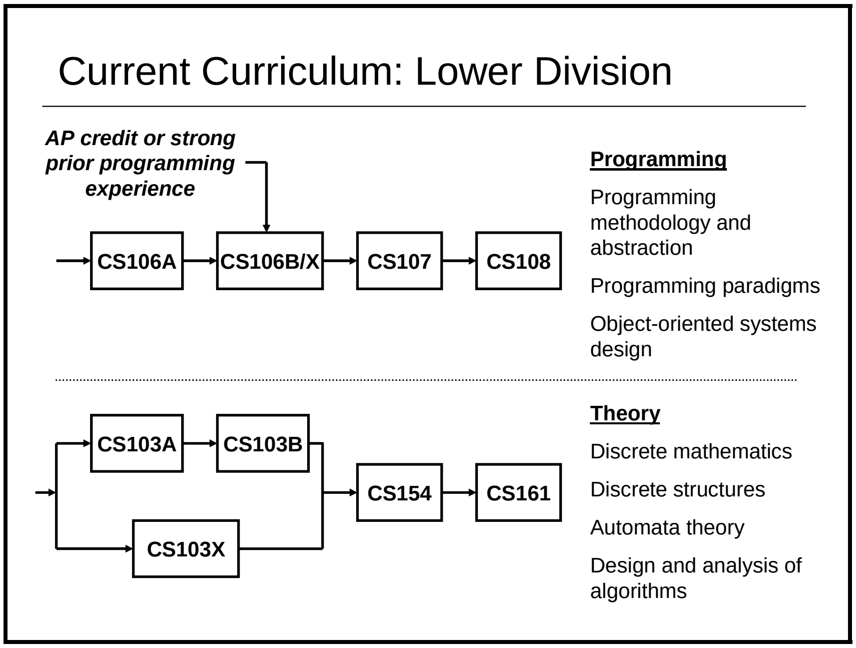
III. Introduction to Theory

- A. Regular Expressions
- B. Finite Automata
- C. Context Free Grammars
- D. Gödel, Turing and Undecidable Problems

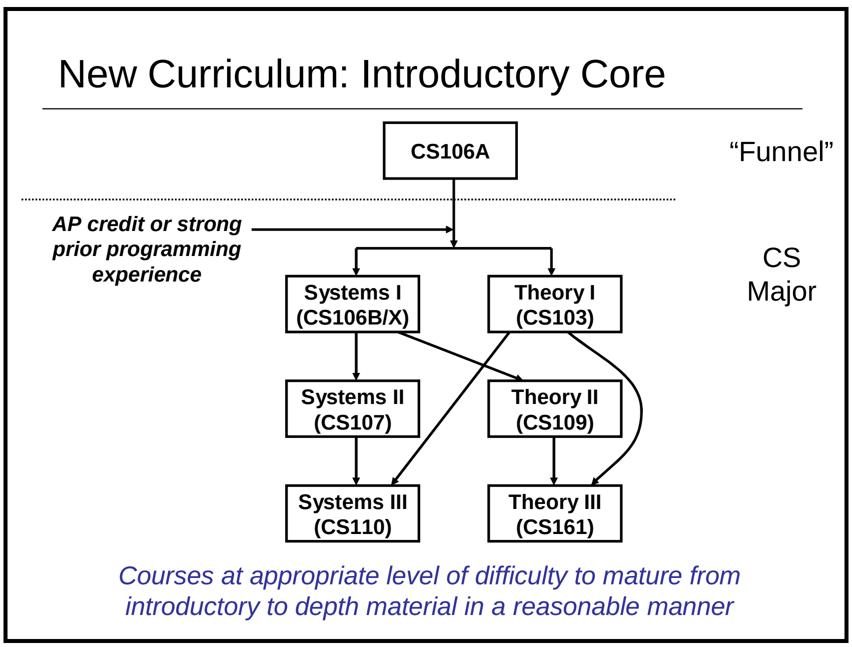
New version of this course next quarter

The CS Department has revised the undergraduate curriculum, which has resulted in some new courses and new degree requirements. You can find slides from a detailed presentation of the restructuring at http://cs.stanford.edu/degrees/undergrad/CurriculumRevision-Preview-04-03-08.pdf .

The current set of core courses link together like this:



Under the new system, the structure will change:



Note that the new CS103 will be a one-quarter course. CS154 will still be taught, but it will not be a general CS requirement. Here, from the Bulletin, are the descriptions of the courses in the new Theory core:

CS 103. Mathematical Foundations of Computing
Mathematical foundations required for computer science, including propositional predicate logic, induction, sets, functions, and relations. Formal language theory, including regular expressions, grammars, finite automata, Turing machines, and NP-completeness. Mathematical rigor, proof techniques, and applications. May not be taken by students who have completed 103A,B or 103X. Prerequisite: 106A or equivalent. GER:DB-Math

CS 109. Introduction to Probability for Computer Scientists
Topics include: counting and combinatorics, random variables, conditional probability, independence, distributions, expectation, point estimation, and limit theorems. Applications of probability in computer science including machine learning and the use of probability in the analysis of algorithms. Prerequisites: 106B or X, and MATH 51 or equivalent. GER:DB-EngrAppSci

CS 161. Design and Analysis of Algorithms
Worst and average case analysis. Recurrences and asymptotics. Efficient algorithms for sorting, searching, and selection. Data structures: binary search trees, heaps, hash tables. Algorithm design techniques: divide-and-conquer, dynamic programming, greedy algorithms, amortized analysis, randomization. Algorithms for fundamental graph problems: minimum-cost spanning tree, connected components, topological sort, and shortest paths. Possible additional topics: network flow, string searching. Prerequisite: 103 or 103B; 109 or STATS 116. GER:DB-EngrAppSci

Here is the plan for the final offerings of the old courses and the initial offerings of the new ones:

Course Transition Plan in 2008-09

Fall	Winter	Spring
CS106X (Systems I)	CS106B (Systems I)	CS106B (Systems I)
CS107 (last old version)		CS107 (new Systems II)
		CS110 (Systems III)
CS103A	CS103B	CS154
	CS103 (Theory I)	CS109 (Theory II)
CS161 (current)	CS161 (current)	

- Existing CS103A/B sequence offered one last time
- Students having taken old CS107 can take CS110
- Content of CS161 will transition (slightly) in 2009-10

The Department will be holding an information session for new and prospective students during the first week or two of this quarter. That will be the place to get more information and get your questions answered.

A Note about the Handouts

This course was originally developed by Maggie Johnson, who still teaches it on occasion. Since the course feeds CS103B, we try to keep it substantially the same each quarter, regardless of the instructor. Though many of the handouts have been created or modified by Robert Plummer and various TA's, we want to take this opportunity to credit Maggie as the original author of most of the handouts that are in prose form (like this one) and to thank her for all her efforts in building this course.