Maggie Johnson Handout #1
CS109

CS109 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 cs_109@hotmail.com. Be sure to monitor the newsgroup frequently: su.class.cs109.

CS109 - Introduction to Computer Science

CS109 is a discrete mathematics course. 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 which 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 problem-solving and algorithmic skills, and to use discrete structures as abstract models for use in solving problems and developing algorithms.

Pre-Requisites

The student is assumed to have a full knowledge of C, and the programming process. Pre-requisites are CS106A and B, or CS106X.

Lectures

Monday, Wednesday, Friday at 11:00 in Terman Auditorium.

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 one midterm, one final, and regular problem sets. All assignments are due <u>in class</u> on the dates specified in the syllabus. You may choose to do the problem sets individually or work in pairs. If you work with a partner, **hand in one copy** of the problem set for both of you. Please do not hand in individual problem sets if you work with a partner - it is very likely a TA will notice the similarity in your work and bring it forward as an honor code issue. It also increases the workload for the TA's. You may work individually on some problem sets and with a partner on others, and you can change partners between problem sets.

If you choose to work with a partner, it is essential that both of you work on the entire problem set together. This is for two reasons: First, we cannot accept "half" of a problem set from you if your partner failed to do his/her part. Second, if you only do half or less of a problem set, there is no way you will pass the exams. The best preparation for the exams is working through the problem sets.

At the beginning of the quarter, each student has 3 "free" class days that can be used as extension days for any assignment. You can use all 3 days on one assignment (you can hand it in 3 class days late) or split them up across two or three assignments. After you use up your three free days, your grade will be docked 10% for each additional late day. Extensions will be granted only if you ask prior to the due date of the problem set, and only if you have already used up your free days. Ask the instructor for extensions, not the TA's.

Make-up examinations will be scheduled only in cases of medical or personal emergency.

Grading

Final grades will be based on the following:

50% Problem Sets
15% Midterm
35% Final

To receive a passing grade, you must complete passing work in both the exam and problem set area.

Textbook

There is a required textbook 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 text book. 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*.

This book and the solutions guide are excellent resources. You will find many examples that can serve as models for the problems on the problem sets.

How to Succeed

The best way to obtain the skills required to succeed in this course is to practice solving problems - lots and lots of problems. The textbook (and the associated web site) provide many practice problems with detailed solutions given in the Solutions Guide. For every problem included on a problem set, it is likely you will be able to find a solved example very similar to the one you are working on. These "model" problems help to illustrate problem-solving thought processes and techniques. The Solutions Guide also has a guide to writing proofs, a list of common mistakes, sample exams, etc. So the way to succeed in CS109: come to class where you will see especially important examples and applications, study the textbook and solutions guide carefully and thoroughly, solve lots of problems, 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

Problem sets are to be done "from scratch", i.e., it is a violation of the honor code to copy or derive solutions from other students, textbooks, or previous instances of this course. Discussion of problem set questions with your partner (only) is allowed. You may not work on problems from the problem sets in a study group. You may, however, work on other problems from the textbook with a study group. Copying of solutions from other students, or from students who previously took this course is not allowed. If you do obtain substantial help from a TA or another student, you must document this in your problem set - this is for your protection. Finally, a good guideline is you must be able to explain and/or duplicate anything that you submit.

CS109 Syllabus

Date	Topic	Due
1/5	Introduction	
1/7	Background & History	Read 1.1, 1.2
1/10	Propositional Logic Problem Set #1 out	Read 1.3
1/12	Introduction to Predicate Logic	
1/14	Resolution	Read 3.1
1/19	Methods of Proof Problem Set #2 out	PS #1 due Read 3.2
1/21	Induction I	

1/24	Induction II	Read 4.1
1/26	Combinatorics I Problem Set #3 out	PS #2 due Read 4.3
1/28	Combinatorics II	
1/31	Program Proofs Recursion	PS #3 due
2/1	7 p.m.- 8:30 p.m. Midterm	
2/2	No class	Read 1.8, 2.2
2/4	Analysis of Algorithms I Problem Set #4 out	Read 5.1
2/7	Analysis of Algorithms II	
2/9	Analysis of Algorithms III	Read 1.4, 1.5
2/11	Infinite Sets and Countability Problem Set #5 out	PS #4 due Read Chap. 6
2/14	Relations I	
2/16	Relations II	Read 8.1, 8.3
2/18	Trees I Problem Set #6 out	PS #5 due
2/23	Trees II	Read 7.1, 7.2
2/25	Graphs I	Read 7.4, 7.5
2/28	Graphs II	
3/1	Graphs III	Read 10.1- 10.4
3/3	Regular Expressions PS #7 out	PS #6 due
3/6	Finite Automata	
3/8	Context Free Grammars	
3/10	Review	PS #7 due

Final: Thurs., March 16, 8:30 -11:30 am

Topic Outline

I. Basic Tools

- 0. Background and History
- A. Overview of Logic
  - 1. Propositional Logic
  - 2. Predicate Logic
- B. Proof Techniques
  - 1. Direct/Indirect
  - 2. Induction
  - 3. Program proofs
- C. Combinatorics
- D. Recursion
- E. Functions
- F. 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. Stacks
  - 3. Queues
- D. Trees
  - 1. Concepts and Definitions
  - 2. Structural Induction
- E. Graphs
  - 1. Concepts and Definitions
  - 2. Paths and Circuits

III. Introduction to Theory

- A. Regular Expressions
- B. Finite Automata
- C. Context Free Grammars