Mathematical Foundations of Computer Science

Ten Powerful Ideas

and applications

CS 15-299 Spring 1998

Main Lecture: TR 10:30-11:50 in 2210 Dougherty Hall
Section A: Mon 10:30-11:20 in 203 Student Center
Section B: Mon 11:30-12:20 in 203 Student Center
Section C: Mon 12:30-1:20 in 203 Student Center
Section D: Mon 1:30-2:20 in 203 Student Center

Professor Maggs	4123 Wean, 268-7654, bmm+@cs
TA (A) Rob Miller	5117 Wean, 268-3046, rcm+@cs
TA (B) Adam Kalai	7113 Wean, 268-7123, akalai+@cs
TA (C & D) Hal Burch	7110 Wean Hall, 268-7670, hburch+@cs
TA Doug Beeferman	5101 Wean, 268-8139, dougb+@cs
Secretary: Dorothy Zaborowski	4116 Wean, 268-3779, daz+@cs

http://www.cs.cmu.edu/afs/cs/academic/class/15299/

Ten Powerful Ideas

Counting: Learn how to count without counting.

Induction: Recognize it in all its guises.

Exemplification: Find a sense in which you can try out a problem or solution on small examples.

Abstraction: Abstract away the inessential features of a problem.

Modularity: Decompose a complex problem into simpler subproblems.

Representation: Understand the relationships between different possible representations of the same information or idea.

Refinement: The best solutions come from a process of repeatedly refining and inventing alternative solutions.

Toolbox: Build up your vocabulary of abstract structures.

Optimization: Understand which improvements are worth it.

Randomization: Flipping a coin can be surprisingly helpful!

Course Schedule

CS 15-299 **Spring 1998**

Lecture	Date	Day	Topic
1	13-Jan	T	Introduction
2	15-Jan	Th	Dating
3	20-Jan	T	Powers
4	22-Jan	Th	Multiplication
5	27-Jan	T	Golden Ratio
6	29-Jan	Th	Design by Induction
7	3-Feb	T	Groups
8	5-Feb	Th	Computer Arithmetic
9	10-Feb	T	Counting
10	12-Feb	Th	Counting
11	17-Feb	T	Counting
12	19-Feb	Th	Counting
	23-Feb	M	**Quiz 1 in Recitation**
13	24-Feb	T	Counting
14	26-Feb	Th	Infinity
15	3-Mar	T	Undecidability
16	5-Mar	Th	Intractability
17	10-Mar	T	Automata Theory
18	12-Mar	Th	Probability
	16-Mar	M	**Quiz 2 in Recitation**
19	17-Mar	T	Probability
20	19-Mar	Th	Probability
	24-Mar	T	Spring Break
	26-Mar	Th	Spring Break
21	31-Mar	T	Probability
22	2-Apr	Th	Probability
23	7-Apr	T	Probability
24	9-Apr	Th	Graph Theory
25	14-Apr	T	Graph Theory
26	16-Apr	Th	Graph Theory
	20-Apr	M	**Quiz 3 in Recitation**
27	21-Apr	T	Graph Theory
28	23-Apr	Th	Cryptography
29	28-Apr	T	Cryptography
30	30-Apr	Th	Surprise

Important Course Information

Bruce Maggs **Jan 12, 1998**

Course Description

This course is about how to rigorously formulate and solve problems relevant to Computer Science. Examples will be taken from Algorithms, Complexity Theory, Game Theory, Probability Theory, Graph Theory, Automata Theory, Algebra, Cryptography, and Combinatorics. Besides providing an introduction to some of the most useful and important ideas from these areas, this class will challenge students to develop their general thinking ability.

The technical content of the course will include an introduction to discrete mathematics: how to count, how to compute probabilities, how to solve sums and recurrences, and graph theory.

Prerequisites

15-127, 21-127, and either 15-211 or concurrent enrollment in 15-211 Lecture II. Special arrangements may be made by permission of the instructor. This course can be substituted for 21-228 in the CS undergraduate curriculum.

Text

There is no required text for the course. The material is fairly diverse, and no standard text contains it. Photocopies of the transparencies used in the lectures will be handed out or made available on the web.

If you want to look at books which contain part of the course material, I recommend the following:

1. Applied Combinatorics, by A. Tucker, published by Wiley & Sons.
2. Elements of Discrete Mathematics, by C.L. Liu, published by McGraw-Hill.
3. Combinatorics: Topics, Techniques, Algorithms, by P. Cameron, published by Cambridge University Press.
4. Concrete Mathematics: A Foundation for Computer Science,by R. Graham, D. Knuth, and O. Patashnik, published by Addison-Wesley.
5. Elementary Probability Theory with Stochastic Processes, by K.L. Chung, published by Springer-Verlag.
6. Fifty Challenging Problems in Probability, by F. Mosteller, published by Dover.
7. Introduction to Algorithms, by T. Cormen, C. Leiserson, and R. Rivest, published by MIT Press.
8. Discrete Mathematics and its Applications, by K. H. Rosen, published by McGraw-Hill.
9. Cryptography: Theory and Practice, by D.R. Stinson, published by CRC Press.

10. How To Solve It: A New Aspect of Mathematical Method, by G. Polya, published by Princeton University Press.

11. Programming Pearls and More Programming Pearls, by J. Bentley, published by Addison-Wesley.

12. Conceptual Blockbusting: A Guide to Better Ideas, by J. L. Adams, published by W. W. Norton & Company.

Web Page, Bboards, and Mailing Lists

The class Web page will provide a wealth of up-to-date information about the course. The URL is

- `http://www.cs.cmu.edu/afs/cs/academic/class/15299/`

If you are using a local UNIX system, a better URL is

- `file:/afs/cs/academic/class/15299/index.html`

It will provide faster access.

From the Web page, you can access the class Bboards. `cmu.andrew.academic.cs.15-299.announce` is for the course staff to make announcements to the class. Please do not post messages on this bulletin board.

`cmu.andrew.academic.cs.15-299.discuss` is for general discussion among members of the class. Please keep discussion relevant and polite.

The Web page will include all assignments and on-line handouts, as well as facilities to check how you are doing in the course. You should visit it frequently.

The class will also have mailing lists: the list cs-299@cs mails to the entire class and teaching staff; cs-299-A@cs, cs-299-B@cs, cs-299-C@cs, and cs-299-D@cs mails only to the specified recitation section (and its TA); cs-299-staff@cs mails only to the teaching staff.

Homework Assignments and Exams

There will be problem sets, programming assignments, three quizzes, and a final exam. The grading formula will be:

35% Homework (problem sets, programming assignments)

30% Quizzes

35% Final

This formula is a lower bound on your grade. If the class average as computed by this formula is lower than a 77, I will use a curve to raise your grades. *Under no circumstances will I use a curve to lower your grades.* From past experience, I expect the formula to determine your grade exactly. 90 and above is A range, 80-89 is B range, 70-79 is C range, and so forth.

Notice that if it is your habit to put in marginal effort on the homework and to come through on the exams, your grade will suffer a bit. This is deliberate; I think the homework is important.

Your mid-term grade will reflect your anticipated final grade.

Typesetting

You must typeset your solutions to the problem sets. This makes the graders' job much easier. I recommend that you learn and use LaTeX, but anything typed is fine. Both Microsoft Word and FrameMaker can typeset equations.

Late Work

The good news is that you can hand in any assignment up to one week late (seven days). The bad news is that you will lose five points (out of 100) per day for the privilege. To be unambiguous, we define a "day" to start at midnight. You will lose five points for each extra midnight that it takes you to do the assignment. *Late work makes a class much harder to administer. It also hurts you. Please try and avoid it.* If your assignment is more than seven days late, you will get a grade of zero (0) for that assignment.

Written assignments can be handed in only once, in their entirety. If you hand in multiple versions of a written assignment, only the first version will be graded.

Programming assignments may be resubmitted any number of times throughout the 7-day grading period. Each night at midnight, starting on the night the assignment is due, our scripts will automatically collect all the new handins and grade them, taking into account the 5-point-per-day lateness penalty. Your best current grade for the assignment will be posted to the Web page by the following morning.

If you slip an assignment under my door, please put the date and time on the top. If I come in late at night and find that you have lied to me, I will consider you to have cheated.

If you have a good excuse (such as being very sick), you should contact me. For compelling reasons (that extend beyond the fact that you have a lot of work lately and didn't plan ahead), I will excuse you from the lateness penalty.

Honors Problems

A few of you will find the assignments too easy. For this reason, I will include honors problems as a more challenging alternative. These problems will be substantially more sophisticated. Of course, you will learn much more by doing them. In each assignment the honors alternative will be made clear. Their point value will be specified.

If you do both the normal problem and the honors version you will get the maximum of the two scores.

If you think that you might want to do research with me, or if you will want a letter of recommendation from me, it is a good strategy to attempt these problems.

Policy on Collaboration and Cheating

You may verbally collaborate on homework problems and the programming assignments. On each problem and program that you hand-in, you must include the names of the people with whom you have had discussion concerning your solution. Indicate whether you gave help, received help, or worked something out together. *You may not share written work or programs with anyone else.* You may not receive help from students who have taken the course in previous years. You may not review any course materials (or software) from previous years. You may not read the current solution (handed out) if you will be handing in the current assignment late. You may <u>not</u> look up the answer to a homework assignment which happens to appear in the published literature. You may <u>not</u> attempt to violate the security of the electronic grading system. You <u>may</u> get help from anyone concerning programming issues which are clearly more general than the specific assignment (e.g., what does a particular error message mean?).

Thus, clear examples of *cheating* include:

- Showing your code to another student.
- Copying a program from someone else.
- Getting help from someone whom you do not acknowledge on your solution.
- Showing a draft of a written solution to another student.
- Copying from another student during an exam.
- Receiving exam related information from a student who has already taken the exam.
- Submitting a program that attempts to alter or erase grading information.
- Looking at someone else's files containing draft solutions, even if the file permissions allow it.
- Lying to the course staff.

Penalty for Cheating

My reaction to your cheating will vary according to the situation.

Unsolicited Confession If you seek me out and admit that you have cheated, I will probably let it go.

Solicited Confession If I come to you and ask if you have cheated and you freely admit it, I will take that into consideration. I will either ask you to drop the class, or fail you in the course.

Denial If you do not admit that you have cheated, I will provide my evidence that you have done so. I will at the very least fail you in the class; furthermore, I will take my evidence to the dean and seek more substantial penalties.

Pedagogical Rationale and Advice

Collaboration not only helps you get the job done, it teaches you how to explain your (inchoate) ideas to others. This is why I permit discussion of the problems between students. Be careful not to let other people do all the work. If you misuse the opportunity for collaboration in this manner, you will fail the exams and do poorly in the course. The best is usually to find a single partner with whom you have a relatively balanced collaboration. A group of four is usually too big for everyone to be following the joint problem solving process.

Many of the course materials will be the same as in previous years. This is not because I am lazy. It takes years to develop good problems. The only reason to change them is to make cheating more difficult. It is far better for you to work on the most excellent problems that I could find in my life so far. I appeal to your sense of honor because this is what is optimal from a pedagogical point of view.

Signing the Statement on the Back Page

I understand that most of you would never consider cheating in any form. There is, however, a small minority of students for whom this is not the case. In the past, when I have caught students cheating they often insisted that they did not understand the rules and penalties. For this reason I require that each student sign and return the back page of this document.

Commitment to Honor the Cheating Policy

Policy on Collaboration and Cheating
You <u>may</u> verbally collaborate on homework problems and the programming assignments. On each problem and program that you hand-in, you <u>must</u> include the names of the people with whom you have had discussion concerning your solution. Indicate whether you gave help, received help, or worked something out together. You may <u>not</u> share written work or programs with anyone else. You may <u>not</u> receive help from students who have taken the course in previous years. You may <u>not</u> review any course materials (or software) from previous years. You may <u>not</u> read the current solution (handed out) if you will be handing in the current assignment late. You may <u>not</u> look up the answer to a homework assignment which happens to appear in the published literature. You may <u>not</u> attempt to violate the security of the electronic grading system. You <u>may</u> get help from anyone concerning programming issues which are clearly more general than the specific assignment (e.g., what does a particular error message mean?).
Thus, clear examples of cheating include:

- Showing your code to another student.
- Copying a program from someone else.
- Getting help from someone whom you do not acknowledge on your solution.
- Showing a draft of a written solution to another student.
- Copying from another student during an exam.
- Receiving exam related information from a student who has already taken the exam.
- Submitting a program that attempts to alter or erase grading information.
- Looking at someone else's files containing draft solutions, even if the file permissions allow it.
- Lying to the course staff.

Penalty for Cheating
My reaction to your cheating will vary according to the situation.

Unsolicited Confession If you seek me out and admit that you have cheated, I will probably let it go.

Solicited Confession If I come to you and ask if you have cheated and you freely admit it, I will take that into consideration. I will either ask you to drop the class, or fail you in the course.

Denial If you do not admit that you have cheated, I will provide my evidence that you have done so. I will at the very least fail you in the class; furthermore, I will take my evidence to the dean and seek more substantial penalties.

Pedagogical Rationale and Advice
Collaboration not only helps you get the job done, it teaches you how to explain your (inchoate) ideas to others. This is why I permit discussion of the problems between students. Be careful not to let other people do all the work. If you misuse the opportunity for collaboration in this manner, you will fail the exams and do poorly in the course. The best is usually to find a single partner with whom you have a relatively balanced collaboration. A group of four is usually too big for everyone to be following the joint problem solving process.
Many of the course materials will be the same as in previous years. This is not because I am lazy. It takes years to develop good problems. The only reason to change them is to make cheating more difficult. It is far better for you to work on the most excellent problems that I could find in my life so far. I appeal to your sense of honor because this is what is optimal from a pedagogical point of view.

I, ______________________________ , have read and understood the
(PRINT YOUR NAME)
above statement of the CS 15-299 policy on collaboration and cheating which was distributed on the first day of class. I agree to honor the rules which the policy statement describes.

____
(SIGN AND DATE)