Syllabus

Instructor

Alan T. Sherman (email: sherman@algol.cs.umbc.edu), Computer Science Department, ECS (Engineering and Computer Science Building) 225J, Tele. (410) 455-2666 [direct], 455-3000 [Department Secretary].

Office hours: Monday and Wednesday immediately after class, and by scheduled appointment. Friday is the best day for scheduled appointments. Please do not disturb Tuesday and Thursday.

Teaching Assistant

William A. Arbaugh (email: arbaugh@super.org). *Office hours: By scheduled appointment—send email to schedule.*

Meeting Time and Place

Monday and Wednesday, 4:00–5:15pm
SS (Social Science) 206

Prerequisites

Calculus I–II (Math 151–152), Linear Algebra (Math 221), and at least one math course more advanced than linear algebra. Or permission of instructor. These prerequisites are the math prerequisites for our M.S. program.

Brief Course Description

Graduate introduction to the fundamental concepts and techniques of discrete mathematics that are essential for the study of computer science. The main goal of this course is to develop mathematical skills and sophistication for proving theorems and solving problems. In addition, this course will help develop skills for modeling problems mathematically, counting, estimating values, and expressing ideas in precise mathematical prose.

Topics include the following. *I. Fundamentals:* sets, elementary logic, proof techniques, problem-solving strategies, numbers, functions and relations, and graphs. *II. Counting:* elementary probability and combinatorics, estimation techniques, Burnside's Lemma. *III. Calculations:* summations, difference equations, and generating functions. *IV. Selected topics:* introductory group theory, elementary number theory, Markov chains, and applications of these topics to computer science.

CMSC–603 and the related undergraduate course CMSC–203 (Discrete Structures) cover similar topics, but at different levels. In particular, 603 places more emphasis on writing proofs. In addition, 603 requires each student to write an original paper; 603 covers some more advanced material (*e.g.* generating functions, Markov chains, Burnside's Lemma); 603 points out more applications of discrete mathematics to computer science; and 603 requires more challenging tests and assignments.

It is highly recommended to take 603 before taking 641 (Algorithms) or 651 (Automata Theory). 603 is an area A course that carries elective credit.

Recommended Texts

Although the course will not closely follow any text, the course material will be based on selected sections from the following two books by Susanna Epp and Fred Roberts. Specifically, for the first part of the course on proof techniques, I recommend Epp. For the second part on combinatorics, I recommend Roberts. In addition, you may find the book by Solow helpful on basic techniques for writing proofs.

The scope of each exam will be limited to the material covered in Epp and Roberts, but at a level similar to problems solved in class and in homework assignments.

1. Epp, Susanna S., *Discrete Math with Applications*, Wadsword (1990). [Undergraduate text at UMBC; one of the better elementary discrete mathematics texts. Good for basic proof techniques and for basic vocabulary of discrete mathematics. Exercises are very elementary.]

2. Roberts, Fred S.; *Applied Combinatorics*, Prentice-Hall (Englewood Cliffs, NJ, 1984). [Good treatment of combinatorics. Does not address the other course topics.]

3. Solow, Daniel, *How to Read and Do Proofs: An Introduction to Mathematical Thought Process*, John Wiley (1990). [A concise helpful guide to writing proofs.]

For your benefit, these and other books on discrete mathematics are on reserve at the UMBC Kuhn Library.[1]

Required Work

Required work consists of weekly homework assignments, ungraded in-class exercises, an original project, two closed-book in-class exams, and a closed-book in-class final exam. In addition, each student is expected to participate actively in class. *The student's most important responsibility is to actively solve problems*, including required homework problems and selected exercises encountered in the readings. In particular, each student should solve at least one or two problems every day. The accompanying schedule lists due dates for all required work.

Each homework assignment consists of required readings and problems to be solved in writing. Students are encouraged to work together, but each solution must be written up individually. Since solving problems is the most effective way to learn, the homework is the most important activity of the course. Most problems will consist of nontrivial proofs and word problems that require creative multi-step solutions. For guidelines and advice on doing homework, see the handout "How to solve and write up homework problems."

For the project, each student will solve a practical computer science problem using discrete mathematics. To carry out this project, each student must complete the following four tasks: 1) Select an important practical problem that arises in computer science. 2) Model the practical problem mathematically, and state an abstraction of the practical problem in a precise, well-defined way. 3) Solve the abstract problem using discrete mathematics. 4) Interpret your

[1] See the handout "Supplemental reading."

solution of the abstract problem in the context of the practical problem. Each project must be original and written up in the form of a brief technical report. The purpose of the project is to apply discrete mathematics to computer science; to provide a flexible, open-ended, creative, exciting learning experience; and to encourage you to think about a nontrivial problem over a long period of time. Detailed information about the project will be distributed in a separate handout.

Each exam is a learning experience that helps the student review and integrate course material. Most exam questions will ask students to apply important concepts to new situations. In contrast with homework assignments, each exam must be solved entirely individually. In addition, since the exams are closed-book, they provide an opportunity to measure how well each student has mastered fundamental facts. To encourage students to solve problems, some exam questions will be drawn directly from unassigned exercises in the required readings. No homework is due during the week of any exam.

Occasionally, students will solve ungraded exercises during class. These exercises help students master important knowledge and skills by actively engaging them in problem solving and theorem proving.

Lectures provide an opportunity for the instructor to motivate and introduce new material, to solve problems, and to answer questions. Students will benefit most from this experience by coming prepared and by actively asking and answering questions. It is strongly recommended that you take minimal lecture notes so that you can focus your energies on actively following each lecture. On a regular basis, students will be asked to solve new problems at the chalk board during class.

Office hours provide an opportunity to obtain individual help and guidance. The most effective way to use office hours is to come with a written list of specific questions. If you have a question about a homework problem, you should also bring in writing your partial work on the problem. The instructor is eager to help and it is important for you to seek out the instructor on a regular basis. But it is inappropriate to expect the instructor to provide extended individual tutorials or to repeat lectures.

Each student is expected to spend a total of approximately 12 hours per week on this course, roughly allocated as follows: 3 hours attending lectures, 30 minutes reviewing lecture notes, 2:25 hours doing assigned reading, 6 hours doing written homework and project (or studying for exams), and 5 minutes seeking individual help from instructor or TA.

Grading Policy

At the end of the semester, each student will receive a term grade as a measure of the student's performance on the required course activities. The term grade will be based on a weighted average of scores from these activities.

For each course activity (*e.g.* homework, tests), the student will receive an activity score. A term score will be computed by taking the weighted sum of the activity scores, using the relative weights given in Table 1. The instructor will convert term scores into letter grades by sorting the term scores for the entire class and by partitioning the sorted list of scores. This partition will be drawn primarily on the basis of absolute performance, without any predetermined cutoff points. The instructor reserves the right to make minor adjustments to the partition based on individual circumstances.

Table 1: Course activities and their relative weights.

	Activity	Weight
1.	Class participation	2%
2.	Written homework	23%
3.	Project	20%
4.	Exam I	15%
5.	Exam II	15%
6.	Final exam	25%
	Total:	100%

In addition, the following grading rules will be applied. To pass the course, the student must make a serious attempt to complete all major activities, including written homework, exams I–II, project, and final exam. In particular, each student is expected to make a serious attempt to solve every required homework problem; any student who does not make a serious attempt to solve at least 60% of the required homework problems will not pass. To earn an 'A' it is necessary (but not sufficient) to make a serious attempt to solve substantially all required homework problems.

Late homework will not be accepted.[2] Late projects will be penalized for each day of lateness and will not be accepted after the last day of class.

Incomplete grades will be issued only under those extreme situations described by University policy for granting incompletes. Failure to complete assignments on time is not a sufficient reason for an incomplete.

The instructor recognizes that occasionally students will have to miss class for legitimate reasons (*e.g.* personal illness). In such cases, it is the student's responsibility to make up all missed activities.

Computing Facilities

Each student should obtain a permanent named account on the UMBC8 computer, which is a Silion Graphics Crimson running the Irix operating system. To obtain an account, go to any computer lab (e.g. in the ECS building) and follow the instructions for applying for an account. Although there will be no programming assignments, students are strongly encouraged to use the computer facilities for email, newgroups, Maple, network access (e.g. mosaic), educational programs (e.g. Tarski's World), word processing, and access to course materials. Some educational programs will be available only through the ECS MacLab.

Students are encouraged to use the C and C++ programming languages, the Gnu Emacs text editor, and the LaTeX document preparation system.

Academic Misconduct

Each student is expected to be familiar with all University and Department policies on academic misconduct. An egregious type of academic misconduct is plagiarism, which in each of its

[2] The main reasons for this policy are to enable the instructor to comment on the homework in class on the day it is due, to encourage each student to keep up-to-date, and to enable the TA to grade all solutions for each assignment in one sitting.

many forms involves representing somebody else's work as your own. Please carefully read the handout "Homework policy" for an explanation of acceptable and unacceptable forms of homework collaboration. Violations of academic honesty will be dealt with severely.