MATH 2358
Discrete Mathematics I

Instructor
Dr. David Snyder

Course Description
A study of discrete mathematical structures that are commonly encountered in computing hardware and software. Prerequisite: MATH 1315 with a grade of "C" of higher, MATH 2417 with a grade of "D" of higher, or MATH 2471 with a grade of "D" of higher.

Scope & Nature of the Course
This course is designed to give computer science majors a sufficient mathematical background to understand the underlying philosophy, logic, and design of digital computers, as well as subsequent issues in software programming. It can also be helpful for those wishing to understand the mathematics of finite systems. Mathematical prerequisites are minimal: a good background in algebra suffices (of course, geometry, calculus, etc. are also helpful). The course can be broken into two parts. The first part focuses on issues directly related to a digital computer's hardware architecture. The second part concentrates on concepts and skills fundamental to software architecture (i.e., programming). Understanding these foundations is essential to anyone wishing to develop a career in any of the numerous areas of computer engineering or science.

This course consists of fifteen lessons. These lessons cover the first four chapters of the course's text. The first broad topic area is the foundation for studying discrete structures: logic and proof. The second topic area is that of sets, functions, and sequences. These topics form the core language needed for a study of modern mathematics and computer science, and their study requires an unerring sense of logic and an understanding of proof. The third topic area focuses on applying the concepts from the first two topic areas to the study of finite procedures, as well as an introduction to analysis of the effectiveness and efficiency of those procedures. The focus is on the set of integers and number theory, where finite procedures abound, with a plentiful supply of simple procedures for beginners to work with. The fourth and final topic area discusses inductive and recursive reasoning, which forms the basis for the structure of most programs and for proving that those programs are correct.

Required Materials
Two texts are required for this course:

> Rosen, Kenneth H. *Discrete Mathematics and Its Applications*. 6th ed. with online access code. New York: McGraw-Hill, 2007. ISBN 978-0-07-2880083
>
> Rosen, Kenneth H., and Jerrold Grossman. *Student's Solutions Guide to Accompany Discrete Mathematics and Its Applications*. 6th ed. New York: McGraw-Hill, 2007. ISBN 978-0-07-3107790

Besides the above texts, you may find it convenient or helpful to own a calculator that performs binary, octal, and hexadecimal conversions and arithmetic. However, no calculators will be allowed during the exams.

Course Goals
Whether you want to start a career as a programmer, systems administrator, database

administrator, systems analyst, etc., there is a common, conventional context and language that computer science and computer engineering professionals share. The goal of this course is to provide you with fundamental concepts, terms, and skills necessary to be conversant in that language. In particular, this first course provides a foundation for the student to learn mathematical reasoning and algorithmic thinking (including introductory-level complexity analysis), as well as be introduced to basic discrete structures such as sets and bit strings. The objectives at the beginning of each lesson provide specific outcomes you must aim for and achieve. After completing the lessons, you will understand the basic mathematical structures used in junior-level (and some sophomore-level) computer science courses.

Course Procedure
Each of the fifteen lessons consists of a reading assignment, a discussion, an independent exercise, and an assignment to be submitted for evaluation. Also, learning objectives are stated for each lesson. Reading the assigned text carefully and working through the examples yourself should lead to your mastery of these goals. Work as many exercises from the text as you can, in addition to the assigned ones (practice makes better). The assignments are designed to give you relevant feedback on your understanding of the lesson and further help you master a lesson's objectives. The exams help me determine how well you have attained those objectives.

Online Resources
The textbook publisher has an online resource site for students. Resources vary by chapter but can include interactive demonstrations, self-assessments, and extra practice problems.

Assignments
This course includes fifteen assignments, which together constitute 20% of the grade for this course. Complete each lesson in this course by submitting the assignment at the end of that lesson.

There are basically three types of exercises that you will turn in for an assignment (though in some assignments, only one type might appear), given below. The minimal amount of work your are required to show is given in each case. In general, show your work, enough of it that the instructor can gauge your understanding of the material. So explain yourself as much as time and space allow.

- **Computational Exercises:** These are basically numerical (hexadecimal addition, for example) or organizational (such as a truth table) in nature; they do not usually require much (if any) explanation.
- **Graphical Exercises:** In these exercises, you have to draw a picture to indicate how you arrive at your result(s). Karnaugh maps and Venn diagrams are examples of such exercises. Depending on the exercise, you may be required to explain and/or justify what the diagram signifies.
- **Expository Exercises:** These are exercises requiring a written response in addition to any computations or pictures you may use. You will be required to explain, justify, or prove your answer.

Required Elements for Assignments

- Assignments must be written on lined notebook paper.
- You must number your pages.
- Your submissions should be neat, legible, and organized; you should, in general, use scratch paper to get your solutions, and then write them up neatly so that your solutions are easy to follow.

- Don't just give an answer; restate questions briefly to ensure you understand the question and put your polished solution underneath.
- Leave room for comments from the instructor.

Only scanned, handwritten submissions will be accepted. That is, students should complete the assigned problems by hand, then scan those pages for submission. Your time is better spent thinking about the exercises and concepts involved than in working to provide solutions in typeset form.

If you do not have a scanner, you can a) check with your public library to see if they have one available for use, b) utilize one of the scanners available at computer labs on the Texas State campus, or c) utilize the scanning services of commercial copy centers such as Staples or FedEx Office. *All scans must be submitted as .pdf files, and students are responsible for ensuring that the .pdf is of sufficient quality for clear printing.*

Exams
There will be four exams, each covering about four lessons. Each exam is worth 20% of your course grade.

An outline of the format, general content, and length of each exam, along with suggestions for your review, is given in the Prep for Exam section at the end of each of the even-numbered lessons. Information on scheduling your exam and arranging for a proctor can be found on the Correspondence Course Information (.pdf) page.

To pass this course, you must meet the following requirements:

- have a passing course average (64% or better),
- submit all assignments,
- have an average score on the four exams of 60% or better, and
- score 50% or better on the fourth exam.

Again, you may not use a calculator on the exams.

Grading Criteria
Your grade for the course will be determined as follows:

Assignments: 20%
Exams: 80%
Total: 100%

A (Excellent): 90–100%
B (Good): 80–89%
C (Average): 70–79%
D (Poor): 64–69%
F (Failure): 63% and below

No pluses or minuses will be added to the final, reported grade.

Again, to pass this course, you must meet the following requirements:

- have a passing course average (64% or better),

- submit all assignments,
- have an average score on the four exams of 60% or better, and
- score 50% or better on the fourth exam.

Free Tutoring Resources
A variety of free tutoring resources are available for students enrolled in correspondence courses. All correspondence students have access to several hours of free online tutoring from Smarthinking for subjects ranging from grammar and writing to mathematics and Spanish. Free online tutoring for writing-related assignments is also available from the Texas State Writing Center. For information on accessing these resources, please visit the Office of Distance and Extended Learning's Free Tutoring page. Currently-enrolled, degree-seeking students able to visit the Texas State campus are eligible for free in-person tutoring from the Student Learning Assistance Center (SLAC) on the fourth floor of Alkek Library and from the Math Lab in Derrick 233.

Faculty-Student Contact
According to "Seven Principles for Good Practice in Undergraduate Education," faculty-student contact is very important. Even though this is a correspondence course, I encourage you to contact me if you have any concerns, questions, or problems. You are welcome to e-mail me by using the Mail tool in the left menu bar. (It is important to keep all mail related to this course contained within this TRACS site.) My policy is that during non-holiday breaks or announced away times, any email I receive between Monday morning and Friday at noon will receive a reply within 48 hours. Emails received between Friday at noon and Sunday night will receive a reply on the next business day.

TRACS Technical Support
Texas State's Information Technology Assistance Center (ITAC) provides phone and LiveChat technical support for TRACS 24 hours a day, seven days a week, 365 days a year. To take advantage of these services, visit ITAC online or call 512.245.ITAC (4822). Note also that a number of online TRACS tutorials are available from TRACS Facts.

Before beginning this online course, it is recommended that you review the minimum hardware and software requirements and other important information available on the ITS Course Information page.

Correspondence Course Information
As a correspondence studies student, it is your responsibility to be familiar with correspondence-related policies and services. To this end, I encourage you to review the Correspondence Course Information (.pdf) page as well as the Correspondence Studies Student Handbook.

Students with Disabilities
The Office of Distance and Extended Learning is committed to helping students with disabilities achieve their educational goals. A disability is not a barrier to correspondence study, and we strive to provide reasonable accommodations to individuals in coursework and test taking. Students who require special accommodations need to provide verification of their disability to the Office of Disability Services, Suite 5-5.1 LBJ Student Center, 512.245.3451 (voice/TTY). Students should then notify the Office of Distance and Extended Learning of any disability-related accommodation needs as soon as possible to avoid a delay in accommodations.

Academic Integrity
The Texas State Academic Honor Code applies to all Texas State students, including

correspondence students. The Honor Code serves as an affirmation that the University demands the highest standard of integrity in all actions related to the academic community.

Final Comments

This course requires diligence and discipline to successfully learn and apply the principles covered. The common thread throughout is mathematical logic. This logic is essential and fundamental for constructing hardware or software that works; in fact, the philosophy underlying the design of the electronic computer is due to mathematicians such as John von Neumann. Learning to think critically about mathematical statements and arguments is very akin to analyzing a program or algorithm for correctness and efficiency. Each exercise will help you refine or discover a logical skill helpful to the modern computer professional.