Overview (Read Me First)

Welcome to CS 2240: Discrete Structures for CS

Discrete Structure is, in all honesty, a math course. However, it forms a critical part of a Computer Science education by forcing us to think analytically and logically. Even though there is not much coding/programming in this course, the skills and knowledge you'll gain here will help you to think about the best way to implement software in the future. Upon successful completion of this course you'll be very well prepared for courses such as Theory of Computation, and Algorithm Analysis. The latter of these two courses is where you will come to see the benefits of this course, since that is when you'll begin to implement more sophisticated algorithms in code. All of that will be much easier if you invest the time in learning the material in this course, so hang on and prepare to challenge yourself.

All of the course information is here in Canvas (e.g. course organization, objectives, program goals, etc.). The Syllabus contains information specific to this section (e.g. textbook, course schedule, etc.), and the Course Grading Policy (Read Me Third) has information about the assignments/tasks and breakdown of points for the overall grade in the course.

If at any time during the semester you have a question, feel free to contact me directly on MS Teams or via email (geinitz@msudenver.edu). In most cases I should get back to you within 24 hours.

I look forward to supporting you as you progress in CS journey.

Good luck!

General Course Info

Course Format

This is an in-person class so we will meet synchronously on Tuesdays and Thursdays from 2pm until 3:50pm. There will generally be 1-2 breaks (5-10 min each) and approximately once a week we will have an in-class quiz that you will complete in pairs. Note that attendance is not strictly required for this course, but midterm and final exams will be done in class (i.e. in person), which means that attendance is required on those two days (the exact dates/times of the exams are still to be determined). Although attendance is not required it is beneficial to be able to come to class to ask questions immediately as they arise, and to interact with your fellow classmates, and to take the in-class quizzes.

Course Learning Objectives

Upon completion of this course you should be able to:

1. Apply formal representations and methods of reasoning to computational problem domains.
2. Explain and construct valid arguments and proofs.
3. Interpret and use basic discrete modeling structures in computational contexts.
4. Apply fundamental computational reasoning skills.

Program Goals and Outcomes

The MSU Denver Computer Science program enables graduates to become:

1. Professionals capable of applying computer science and software engineering principles and practices.
2. Graduate students in computer science and related fields.
3. Life-long learners capable of self-study, continuing education and ongoing professional development.
4. Ethical practitioners in computer science, software engineering and related fields.
5. Innovators able to respond to technological change and intellectual challenge.

The program accomplishes the above by providing students with the following (upon graduation):

1. An ability to apply knowledge of computing and mathematics appropriate to the discipline.
2. An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution.
3. An ability to design, implement and evaluate a computer-based system, process, component, or program to meet desired needs.
4. An ability to function effectively on teams to accomplish a common goal.
5. An understanding of professional, ethical, legal, security, and social issues and responsibilities.
6. An ability to communicate effectively with a range of audiences.
7. An ability to analyze the local and global impact of computing on individuals, organizations and society.
8. Recognition of the need for, and an ability to engage in, continuing professional development.
9. An ability to use current techniques, skills, and tools necessary for computing practices.
10. An ability to apply mathematical foundations, algorithmic principles, and

computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices.

11. An ability to apply design and development principles in the construction of software systems of varying complexity.

Course Syllabus

CS 2240: Discrete Structures for Computer Science

- **Location:** AES 220
- **Days:** Tuesdays and Thursdays
- **Time:** 2:00pm - 3:50pm
- **MS Teams Code:** j8txqe0

Course Description

This course provides a solid theoretical foundation for the understanding of computer science, with emphasis on the application of formal structures and reasoning to problems in computer science. The course introduces and demonstrates application of discrete mathematics concepts commonly used in computer science and needed to solve many computational problems. Topics include formal logic systems, Boolean algebra, techniques for formal reasoning (including proof methods), set theory, graph theory, functions, relations, sequences, and recursive structures.

Prerequisites

([(MTH 1110 or equivalent) AND MTH 1120] OR MTH 1400) AND CS 2050

Credit Hours

A credit hour is an amount of work represented in intended learning outcomes and verified by evidence of student achievement that is an institutionally-established equivalency that reasonably approximates not less than one hour of instruction and a minimum of two hours of out-of-class student work each week for approximately fifteen weeks for one semester.

This equates to 45 hours per credit per semester, or 180 hours for this class.

Instructor

- Name: Steve Geinitz
- E-mail: geinitz@msudenver.edu
- Office Hours: see Instructor Introduction
- Office: AES 200X

Important Dates

Deadlines to drop the course are Sunday, August 28th (100% refund), and Wednesday, September 7th (50% refund). November 4th is the deadline to withdraw from a course.

Other Important Course Info

- Overview (Read Me First)
- Course Grading Policy (Read Me Third)

Materials/Resources

The course utilizes several freely available texts. Listing these in roughly the order of their usage in our course, the first is, "An Active Introduction to Discrete Mathematics and Algorithms", by C. Cusack and D. Santos (ver. 2.7.2). The next is, "Discrete Structures for Computer Science: Counting, Recursion, and Probability", by M. Smid. The final resource is, "Applied Discrete Structures", by A. Doerr and K. Levasseur. In general, it should be

possible to successfully complete course using only the information provided during lectures, quizzes, and assignments. However, following along in the texts can be helpful for mastery of the material and so relevant sections from the texts will be given as additional and optional reading.

Although there is not a lot of programming required in this class we will write some in order to better understand the process of going from pure logical ideas to code implementation. We will primarily use will Python for this. Python is a general-purpose language that was designed to be easy to learn. As such, it is widely used and consistently ranks as one of the top programming languages sought by employers. Don't worry if you have not used Python before as we will spend some time get acquainted with it. There are also plenty of resources available to learn the basics of Python (lessons, videos).

Schedule

Below is an "at-a-glance" view of the course topics, readings, and activities, etc. we will cover. Note, however, that the dates below are tentative and may shift as the semester progresses.

Date	Week	Lecture	Topic
Aug 23	1	1	Course Intro CS 1 & CS 2 Review
Aug 25	1	2	Proof Methods
Aug 30	2	3	Proof Methods
Sep 1	2	4	Proof Methods
Sep 6	3	5	Proof Methods
Sep 8	3	6	Logic

Sep 13	4	7	Logic
Sep 15	4	8	Logic
Sep 20	5	9	Sets, Functions, Relations
Sep 22	5	10	Sets, Functions, Relations
Sep 27	6	11	Sets, Functions, Relations
Sep 29	6	12	Sequences and Summations
Oct 4	7	13	Sequences and Summations
Oct 6	7	14	Algorithm Analysis
Oct 11	8	15	Algorithm Analysis Review for Midterm
Oct 13	8	16	Midterm
Oct 18	9	17	Algorithm Analysis
Oct 20	9	18	Algorithm Analysis
Oct 25	10	19	Recursion and Induction
Oct 27	10	20	Recursion and Induction
Nov 1	11	21	Recursion and Induction
Nov 3	11	22	Recursion and Induction
Nov 8	12	23	Combinatorics
Nov 10	12	24	Combinatorics
Nov 15	13	25	Combinatorics
Nov 17	13	26	Graph Theory
Nov 21 - 25			Thanksgiving Break
Nov 29	14	27	Graph Theory
Dec 1	14	28	Graph Theory
Dec 6	15	29	Graph Theory
Dec 8	15	30	Review for Final
Finals Week	16		Final Exam: TBD

Dec 12 - 17			

Course Grading Policy (Read Me Third)

Your final grade for the course will be based on assigned homework, exams, participation, and quizzes. An approximate breakdown and description of each is below. Note that this is approximate and that the final breakdown may fluctuate slightly.

Assessment Type	Number of Items	Proportion of Final Grade
Programming Assignments/Activities	5-15	30%
Participation	2-3	10%
Quizzes	16-20	10%
Midterm	1	25%
Final	1	25%
	Total	**100%**

Homework Assignments

The ability to implement logical concepts and steps into code that runs correctly is an important skill to have as a computer scientist that understands how to think mathematically. These programming assignments/activities will be small in nature but will require careful thinking to be able implement them. We will use www.leetcode.com to do these assignments, which will also help you in the future to gain the skills and confidence needed to pass technical interviews once you begin job hunting. For our class though, all is required is that you complete the problem that is assigned, and that it appears on your leetcode profile. There will be more details coming during class. Although you can collaborate with others on

these small assignments/activities, it is best if you attempt to tackle them on your own. Quizzes and exams may have questions related to these problems so you must understand the problem, which is difficult to do if you do not obtain a solution on your own.

Participation

There will be a few online discussion assignments given, and other possible activities related to the coursework. There will generally not be a right or wrong answer for these, so merely making an attempt should earn full credit.

Quizzes

Despite the relatively small contribution to the overall grade quizzes are an important part of this course. The quizzes will focus on the most recent material, but can also include content from earlier in the semester. Quizzes will come in pairs (i.e. an "A quiz" and a "B quiz") with the first one being done on your own outside of class, and the second being done jointly with a partner during class.

Exams

The midterm and final exams will be comprehensive and will therefore cover all content that has been discussed up to the date of the exam. The exact format and length will be given prior the exam date.

Late Work Policy

Homework assignments/activities will generally be due around 1 week after

being assigned. There will often be a few moments at the end of each class to ask questions about, or even work on an assignment. Thus, there should be ample time opportunities to start right away, which is highly recommended so that you have time to complete it. Late assignments will generally not be accepted unless discussed prior to the due date.

Grading Scale

The points earned on each assessment are totaled and the final grade is determined according to the following table:

Letter grade	Percentage Points
A	90 - 100
B	80 - 89
C	70 - 79
D	60 - 69
F	0 - 59

Tracking Your Learning Progress

At any point during the semester you may click the **Grades** link on the course navigation menu to view your grades and access specific feedback.