CMPSC 360
Discrete Mathematics for Computer Science
Required Course in Computer Science and Computer Engineering

Catalog Data: Discrete Mathematics for Computer Science (3)
Discrete mathematics and foundations for modern computer science. Topics include sets, relations, logic, algorithms, graphs, finite state machines and regular expressions. Prerequisite or Concurrent: CMPSC 122.

Typical Textbook: Kenneth H. Rosen, Discrete Mathematics and its Applications, 4th Edition, McGraw-Hill, 1999
or
Susanna S. Epp, Discrete Mathematics with Applications, 3rd Edition, Brooks/Cole Publ. Co., 2004

Course Objectives: In this course, students learn about the basic objects of discrete mathematics, like sets, relations, functions and graphs, as well as about discrete structures with special relevance to computing, like finite state machines and regular expressions. They learn about logic and proofs (in particular proofs by induction) in order to enable precise mathematical reasoning in the context of discrete structures. The main goals are to lay the foundation for the study of data structures and the analysis of algorithms, and to teach students how to think mathematically. Some proofs might involve properties of algorithms.

Primary Course Outcomes: Upon completion of the course, students should be able to successfully complete:

- Explain definitions of basic concepts of discrete mathematics.
- Correctly use boolean operations.
- Use variables and quantifiers properly. This includes an understanding of the concept of binding a variable. It requires the ability to write a formula expressing a statement given in everyday language, as well as the ability to comprehend the meaning of a given formula.
- Use truth tables to decide whether a simple propositional formula is a tautology.
- Employ indirect proofs.
- Prove simple facts by induction, including strong induction.
- Understand recursive definitions and recursive algorithms.
- Solve recurrence relations.
- Recognize special relations like partial orderings or equivalence relations.
- Know the basic terminology of trees.
- Use O-notation for simple examples.
- Determine the time complexity of some very simple algorithms.
- Argue about the correctness of recursive programs.

Relationship to Undergraduate Program Outcomes: CMPSC 360 is providing the foundation for analyzing the space/time complexity of algorithms using discrete mathematics, including the appropriate use of O-notation and recurrence relations, and CMPSC 360 teaches proof methods that can be used to analyze algorithms and computer code for correctness and efficiency.

Required Topics: Logic: boolean operations and quantifiers
Sets, relations, functions
Growth of functions, the O-notation
Algorithms, integers, matrices
Indirect proofs, proofs by induction including strong induction
Proofs of properties of algorithms
Recursive definitions, recursive algorithms
Permutations, binomial coefficients

	The basics of probability theory Relations and their properties Trees
Class Format:	Three lectures per week.
Professional Component:	CMPSC 360 provides a discrete mathematics foundation for future courses and professional work of computer engineers and scientists. It encourages precise reasoning, which will always be useful.
Evaluation:	40 – 60 % Homeworks and Quizzes 40 – 60 % Exams 0 – 10 % Attendance

Author: Piotr Berman, Martin Fürer, Jonathan Goldstine, Adam Smith, Sofya Raskhodnikova
Last Revised: June 1, 2007