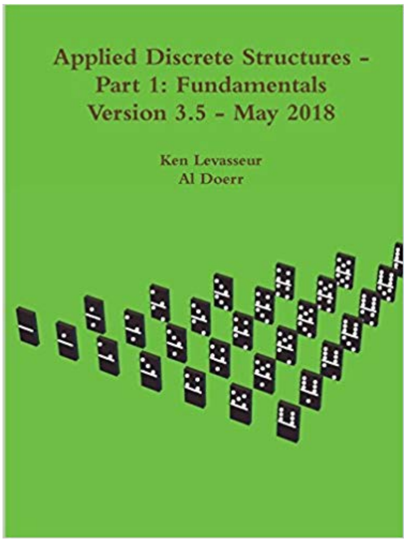
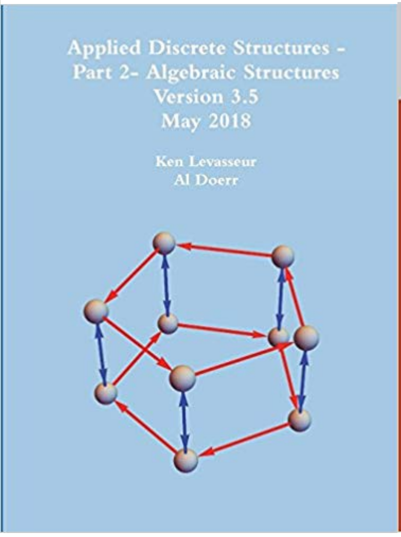
CSCI 006 Discrete Mathematics for Computer Science

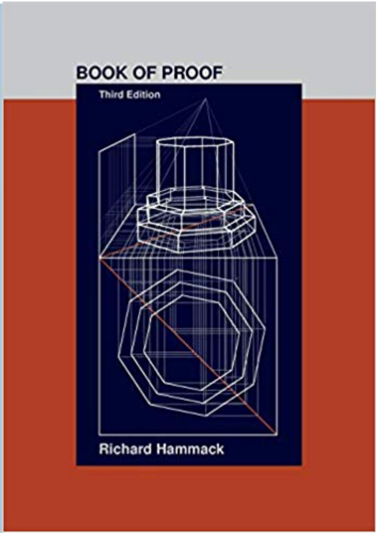
Section:	26298, Spring 2019	Lec / Lab	TR 4:45-6:00 PM
Room:	Alpine 3	Instructor:	Bradley (R B) Powers
		Email:	bradp@cos.edu

Textbooks: **Applied Discrete Structures**, Version 3.5, 2018, by Al Doeer & Ken Levasseur
http://faculty.uml.edu/klevasseur/ads2/
Book of Proof, 3rd Edition, 2018, by Richard Hammack
http://www.people.vc.com/u.edu/~rhammack/BookOfProof/

The texts are open source textbooks in the public domain. Please download a PDF copy from the internet; a hardcopy version is available for purchase from Amazon and others, but is not required







Prerequisites: CSCI 001 or equivalent college course with "C" or better

General Course Description

This course is an introduction to the discrete structures used in Computer Science with an emphasis on their applications. Topics covered include: Functions, Relations and Sets; Basic Logic; Proof Techniques; Basics of Counting; Graphs and Trees; and Discrete Probability.

Course Content

I. Functions, Relations and Sets
   1. Functions (surjections, injections, inverses, composition)
   2. Relations (reflexivity, symmetry, transitivity, equivalence relations)
   3. Sets (Venn diagrams, complements, Cartesian products, power sets)
   4. Pigeonhole principles
   5. Cardinality and countability

II. Basic Logic

1. Propositional logic
2. Logical connectives
3. Truth tables
4. Normal forms (conjunctive and disjunctive)
5. Validity
6. Predicate logic
7. Universal and existential quantification
8. Modus ponens and modus tollens
9. Limitations of predicate logic

III. Proof Techniques
1. Notions of implication, converse, inverse, contrapositive, negation, and contradiction
2. The structure of mathematical proofs
3. Direct proofs
4. Proof by counterexample
5. Proof by contradiction
6. Mathematical induction
7. Strong induction
8. Recursive mathematical definitions
9. Well orderings

IV. Basics of Counting
1. Counting arguments
2. Sum and product rule
3. Inclusion-exclusion principle
4. Arithmetic and geometric progressions
5. Fibonacci numbers
6. The pigeonhole principle
7. Permutations and combinations
8. Basic definitions
9. Pascal’s identity
10. The binomial theorem
11. Solving recurrence relations
12. Common examples
13. The Master theorem

V. Graphs and Trees
1. Trees
2. Undirected graphs
3. Directed graphs
4. Spanning trees/forests
5. Traversal strategies

VI. Discrete Probability
1. Finite probability space, probability measure, events
2. Conditional probability, independence, Bayes’ theorem
3. Integer random variables, expectation

4. Law of large numbers

Course Objectives
At the conclusion of this course, the student should be able to:

1. Describe how formal tools of symbolic logic are used to model real-life situations, including those arising in computing contexts such as program correctness, database queries, and algorithms.
2. Relate the ideas of mathematical induction to recursion and recursively defined structures.
3. Analyze a problem to create relevant recurrence equations.
4. Demonstrate different traversal methods for trees and graphs.
5. Apply the binomial theorem to independent events and Bayes' theorem to dependent events

Methods of Evaluation
Exams
Quizzes
Programming Projects
Discussions
Class Presentations

Student Learning Outcomes

1. Given a proposition, provide a proof presented with clear and correct predicate and propositional logic statements
2. Given a suitable algorithm or theorem, verify the correctness of the algorithm or the theorem by writing a proof using mathematical induction with clear and correct statements
3. Given a predefined relation, determine whether it is reflexive, transitive, or symmetric. Provide an argument using clear and correct proof statements.
4. Given a programming task of searching or sorting items, generate a tree as subset of a graph and construct an efficient algorithm based on structure of the tree
5. Given a suitable algorithm or program, use recursion to analyze the algorithm or program
6. Use sets to solve problems in combinatorics and probability theory
7. Given a graph or tree, apply matrices to analyze the graph or tree
8. Use finite state machines to model computer operation

Important Dates

First Day of Class:	Jan 14, 2020
Last Day to Drop w/o “W”	Jan 27, 2020
Census Date	Feb 3, 2020
Last Day to Drop with a “W”	Mar 20, 2020
Spring Break, No Class	Apr 6 – Apr 10, 2020
Final Exam	May 19, 2020 (4:10 – 6pm)

Class Grading

Unit Assignments and Quizzes	50% of your grade
3 Midterms (the lowest will be replaced by final)	30% of your grade
Final Exam	20% of your grade

Final

The Final will be on May 19, 2020 from 4:10 to 6pm. The final is worth 20% of your grade. The final will also replace your lowest test score if it is higher than your lowest test score.

Grade Scale

90% - 100%	A
80% - 90%	B
70% - 80%	C
60% - 70%	D
0% - 60%	F

Homework

The highest grade a student may earn on late homework is 50% of the points possible.

Canvas Learning Management System

We are using the COS Canvas system for our course management. Please login to Canvas regularly; It contains the course schedule and the homework assignments. I will be posting new course material, fresh assignments, announcements and schedule changes throughout the term. You are responsible for checking Canvas and knowing when work is due.

- Electronic Work Submission: You must use Canvas to turn-in your coursework. You will upload your work as electronic files in MS Word, PDF, or other comparable format. If you need to handwrite your work on paper, particularly to write down the special math symbols, then you must scan the resulting manual pages into a PDF file. All work must be in an electronic format and go through Canvas.