CSC/MAT-220: Discrete Structures

Fall 2017

Science is what we understand well enough to explain to a computer. Art is everything else we do.

Donald Knuth

Instructor:	Thomas R. Cameron	**Time:**	M,W,F 1:30 – 2:20 pm
Email:	thcameron@davidson.edu	**Place:**	WALL 320 (classroom)
			WALL 380 (lab)

Course Page: `https://www.thomasrcameron.com/pages/MAT-220/mat_220.php`
Office Hours: M,W,F 2:30 – 4:00 pm, Tu 10:30 am – 12:00 pm, and by appointment in CHAM 3044.

Textbook: Scheinerman, *Mathematics: A Discrete Introduction*, (3e), 2013
Technology: Standard ML ()
Prerequisite: MAT 113 or equivalent and the ability to program in a high-level language like Python, C++, or Java at the level expected in CSC 121 or an equivalent course.

Course Description: Discrete Structures is an introduction to proof techniques, with a focus on topics relevant to computer science. Topics include: fundamental proof techniques, boolean logic, sequences and summations, set theory, algorithm analysis, recursion, mathematical induction, recurrence relations, an introduction to number theory, combinatorics, discrete probability, and graph theory. The class will be adequate preparation for students choosing to continue on the pure math track (Real Analysis, Abstract Algebra, etc.) or the theoretical computer science track (Analysis of Algorithms, Theory of Computation, etc.).

Learning Outcomes: Students will be able to

- Sets, Relations, and Functions
  - Explain with examples the basic terminology of functions, relations, and sets.
  - Perform the operations associated with sets, functions, and relations.
  - Relate practical examples to the appropriate set, function, or relation model, and interpret the associated operations and terminology in context.
- Basic Logic
  - Convert logical statements from informal language to propositional and predicate logic expressions.
  - Apply formal methods of symbolic propositional and predicate logic, such as calculating validity of formulae and computing normal forms.
  - Use the rules of inference to construct proofs in propositional and predicate logic.
  - Apply formal logic proofs and/or informal, but rigorous, logical reasoning to real problems, such as predicting the behavior of software or solving problems such as puzzles.
  - Describe the strengths and limitations of propositional and predicate logic.
- Proof Techniques
  - Identify the proof technique used in a given proof.
  - Outline the basic structure of each proof technique (direct proof, proof by contradiction, and induction) described in this unit.

  - Apply each of the proof techniques (direct proof, proof by contradiction, and induction) correctly in the construction of a sound argument.
  - Determine which type of proof is best for a given problem.
  - Explain the parallels between ideas of mathematical and/or structural induction to recursion and recursively defined structures.
  - Explain the relationship between weak and strong induction and give examples of the appropriate use of each.
  - State the well-ordering principle and its relationship to mathematical induction.

- Basics of Counting
  - Apply counting arguments, including sum and product rules, inclusion-exclusion principle and arithmetic/geometric progressions.
  - Apply the pigeonhole principle in the context of a formal proof.
  - Compute permutations and combinations of a set, and interpret the meaning in the context of the particular application.
  - Map real-world applications to appropriate counting formalisms, such as determining the number of ways to arrange people around a table, subject to constraints on the seating arrangement, or the number of ways to determine certain hands in cards (eg, a full house).
  - Solve a variety of basic recurrence relations.
  - Analyze a problem to determine underlying recurrence relations.
  - Perform computations involving modular arithmetic.

- Graphs and Trees
  - Illustrate by example the basic terminology of graph theory, and some of the properties and special cases of each type of graph/tree.
  - Demonstrate different traversal methods for trees and graphs, including pre, post, and in-order traversal of trees.
  - Model a variety of real-world problems in computer science using appropriate forms of graphs and trees, such as representing a network topology or the organization of a hierarchical file system.
  - Show how concepts from graphs and trees appear in data structures, algorithms, proof techniques (structural induction), and counting.

- Discrete Probability
  - Calculate probabilities of events and expectations of random variables for elementary problems such as games of chance.
  - Differentiate between dependent and independent events.
  - Identify a case of the binomial distribution and compute a probability using that distribution.
  - Apply Bayes theorem to determine conditional probabilities in a problem.
  - Apply the tools of probability to solve problems such as the average case analysis of algorithms or analyzing hashing.
  - Compute the variance for a given probability distribution.
  - Explain how events that are independent can be conditionally dependent (and vice-versa) Identify real-world examples of such cases.

Grading Policy:
Your final grade is broken up as follows.

Category	Percentage
Lab	10%
EFY	15%
Homework	25%
Midterm Review	25%
Final Review	25%

Your final letter grade is based on the following scale.

Grade	Percentage Interval	Grade	Percentage Interval
A	$[93, 100]$	C+	$[76, 80)$
A-	$[90, 93)$	C	$[73, 76)$
B+	$[86, 90)$	C-	$[70, 73)$
B	$[83, 86)$	D+	$[66, 70)$
B-	$[80, 83)$	D	$[63, 66)$
		F	$[0, 63)$

Lab: Starting the second week in the semester, we will meet every Friday in the lab room to explore programming topics using SML. Lab assignments will be posted online in advance. Students are expected to work through the lab examples, complete the assignment in your own SML file, and email that file with your name and assignment number to me by the following Monday. You may work with anyone you wish and even use online resources to complete your lab, but turn in your own work.

EFY: Exercises For You will be given during each class period. Students will work on these exercises in groups, but turn in their own work. Each EFY can be turned in that day or the following class period. I will drop your three lowest scores.

Homework: Every other week homework assignments will be due that include a large range of problems that will test the students ability to prove theorems, solve problems, and think abstractly. These assignments are posted online two weeks in advance, and are due in class. Students are expected to engage in conversational collaboration, but should not copy eachothers work.

Midterm: On October 6 students will be given a midterm that will test their understanding of the concepts covered up to that point. Students will work on the midterm independently at their leisure over the week, for no more than 3.5 hours, and turn it in during class on October 13.

Final Review: The final will test the students on a comprehensive selection of topics from the course. The test will be administered via Davidson's self scheduled exam system.

Academic Honesty: Students are expected to complete all graded work in accordance with the Davidson College Honor Code, as it applied to each assignment in this class.

Special Accommodations: Davidson College values the diversity of its community and is an equal access institution that admits otherwise qualified applicants without regard to disability. The college seeks to accommodate requests for accommodations related to disability that are determined to be reasonable and do not compromise the integrity of a program or curriculum. To make such a request or to begin a conversation about a possible request, please contact Beth Bleil, Director of Academic Access and Disability Resources, in the Center for Teaching and Learning by visiting her office in the E.H. Little Library, by emailing her at bebleil@davidson.edu, or by calling 704-894-2129. It is best to submit accommodation requests within the drop/add period; however, requests can be made at any time in the semester. Please keep in mind that accommodations are not retroactive.

Disclaimer: I reserve the right to diverge from this syllabus in the best interest of the course. Any changes made will be announced.