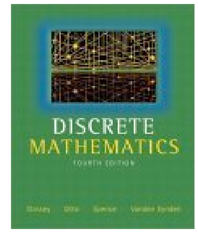
Syllabus for

Foundations of Computing Theory

Computer Science **COMP 111**

Instructor:	Mark LeBlanc (`mleblanc`)	Office Hours: by appt. *or*
Office:	SC-B103	MW 9:30-10:30, 3:30-4:30; F 9:30-10:30
Phone:	286-3970 (on campus: x3970)	Meeting: MWF 10:30-11:20

Required Text:
Discrete Mathematics (4th Ed.) by Dossey, Otto, Spence, and Vanden Eynden. Addison-Wesley, Boston, MA, 2002.



Supplement:
Computer Science Illuminated by Dale and Lewis. Jones and Bartlett Publishing, Boston, MA, 2002.

Content:

Discrete mathematics represents the language, symbolic notation, and problem solving principles that lead to a rich appreciation of computing. This course is an initial semester of exposure to the tools for precise vocabulary, powerful notation, useful abstractions, and rigorous thinking that are needed as someone works in computing. And just who does not work with computing these days in one way or another? A working premise of the course is that it is not possible to make excellent and effective use of computers without involving oneself in mathematical considerations. It seems everyone these days wants to apply computers to the problem at hand, but very few have experience with the fundamental mathematical principles to ensure that things are done correctly and efficiently. Simply put, someone in your group has to know with certainty that an answer is wrong or that a task could be performed more efficiently! This course provides practice with some of the mathematics that enables *you* to be that person.

> *"As the field of computer science matures, more and more sophisticated analysis techniques are being brought to bear on practical problems. To understand the computational techniques of the future, today's students will need a strong background in discrete structures."* (Computing Curricula 2001).

Curriculum: Many areas of computing require an ability to work with discrete mathematical structures. Most of the material covered in this course serves as an initial exposure to and practice with the discrete mathematical topics that appear in later computer science courses. In addition to satisfying the Mathematics/Logic (ML) or Quantitative Analysis (QA) general education requirement, this course can count as the mathematics course required for a computer science minor or as one of the three mathematics courses that are required for a computer science major. A computer science major will see additional discrete math in the required MATH 211 that provides further work in these areas including writing proofs, counting, and graph theory.

Your grade:	In class participation	6%	attendance and participation required
	10 Homeworks	50%	continual throughout the semester
	Exam1	12%	Friday, March 5, in class
	Exam2	12%	Friday, April 23, in class
	Final Exam	20%	Wednesday, May 12, 9am

"In computer science, if you are almost correct you are a liability."
Fred Kollett (1941-1997), Math/CS, Wheaton College

Week	Open Questions	Reading Homework Exams	Topics
1			**Computers and Discrete Math**
Jan 28	How long will it take our group to ship this software? And what is the critical path of tasks that could hold it up?	Dossy et al. 1.1	Critical path analysis
Jan 30	How many possible ways can I burn songs on this CD? (of course, these are legal copies of songs)	1.2 HW1 due Mon, Feb 2	Combinatorics Existence, Counting, and Optimization
2			**Sets, Relations, and Databases**
Feb 2	How can we use congruence to help us detect errors in textbook ISBN numbers?	Dossy et al. 2.1, 2.2, 2.3	Set Operations Sequences and Strings Equivalence Relations
Feb 4	Hey, the relational database model is based on set theory and first order predicate logic, right?	Appendix B Dale Ch. 12	Congruence Matrices of Relations **Relational Databases**
Feb 6	How can we leverage this math to help us design efficient databases?	HW2 due Fri, Feb 6	
3			**Graphs**
Feb 9	How can we store our huge graph in the computer?	Dossy et al. Appendix B 3.1, 3.2, 3.3	Notation Matrices of Graphs Paths and Circuits
Feb 11	What is the shortest path between cell towers to transmit a wireless message across the country?	HW3 due	Data Structures for Graphs Adjacency Matrix and Adjacency List Shortest-path, Breadth-First
Feb 13	How can we visit all nodes on the graph?	Fri, Feb 13	
4			**Trees**
Feb 16	So we know that graphs that are connected and have no cycles are Trees...	Dossy et al. 4.1 4.2	Notation Spanning Trees
Feb 18	How can we help seven farms in Iowa build a communications network to relay storm information with the minimum number of expensive fiber optic lines?	4.3 4.4, 4.5, 4.6 Notes	Depth-First-Search Binary Trees Game Trees
Feb 20	How many moves should my computer game "look ahead" when playing in expert-mode?	HW4 due Fri, Feb 20	
5			**Logic**
Feb 23	How can we describe this situation with propositional statements?	Dossy et al. Appendix A.1	Statements Equivalence Negation with quantifiers
Feb 25	How can we use boolean algebra to find design flaws in our software?	A.2	Tarsky's World **Formal Methods**
Feb 27	How should we document our functions so that others can understand our software?	HW5 due Fri, Feb 27	conditionals PRE/POST conditions Loop invariants

Week	Open Questions	Reading	Topic
6			**Number systems**
Mar 1	So red is `0xFF0000`, right?	Handouts and notes	Binary Octal Hexidecimal
Mar 3	Hey, what is this `35BCF4F` in this error message?		Overflow, Round-off error
Mar 5	What is the largest possible value I can store in a memory location on this chip?	Exam I Mar. 5	**Exam I**
7			**Circuits**
Mar 8	How do tiny embedded microprocessors control larger machines based on a set of inputs?	Dossy et al. 9.1 9.2	Logic gates Boolean algebra
Mar 10			
Mar 12	Just how do those vending machines work anyway?	9.4	**Finite State Machines**
8	**SPRING BREAK**	**SPRING BREAK**	**SPRING BREAK**
9			**Recursion**
Mar 22	How do we mathematically express a "divide and conquer" problem solving strategy?	Dossy et al. 8.1, 2.6	Counting revisited Recurrence relations
Mar 24	If we use our recursive algorithm, how many arithmetic operations will be required?	Notes	**Proof by Induction**
Mar 26		HW6 due Fri, Mar 26	
10			**Languages and Grammars**
Mar 29	What is the syntax for a legal variable name in our programming language?	Handouts and notes	Chomsky hierarchy Context-free grammars, BNF
Mar 31	So why do I need a compiler?		Lexical analysis
Apr 2	What is XML and why is it important?	HW7 due Fri, Apr 2	XML
11			**Languages and Grammars**
Apr 5	What is a regular expression?	Notes	Regular Expressions (Regex)
Apr 7	Will [AG].{3}GC match `GTATGC`?	Travels in DNALand	Regex meets Genomics
Apr 9	Where are the regulatory motifs in DNA sequences?	HW8 due Fri, Apr 9	Regex and Perl
12			now for something continuously different … **Differential Calculus**
Apr 12 Apr 14	So we know our program must deal with *really* large number of data items, how can we compare the rates of growth of two algorithms?	Handouts and notes	Functions Rates of growth Algorithm efficiency Algorithm analysis, "Big Oh"
Apr 16	How can we fit multiple cubic and quadratic polynomials together to approximate a data set?	HW9 due Fri, Apr 16	Spline curves

Week	Open Questions	Reading	Topic
13			**Matrices** revisited
Apr 19	How much do we spend on coffee and candy a day?	Handouts and notes	Matrix operations
Apr 21	How can we store the data points for a cube?		Representing and moving objects in 2-space
Apr 23	How can we rotate the cube?	Exam II Fri, Apr 23	**Exam II**
14			**Experimentation and the Scientific Method**
Apr 26	Can we *really* abstractly represent the runtime of an algorithm by determining the number of steps it requires?	Notes and handouts	Hypotheses Experimental procedure Experimental error
Apr 28	How should we take care to avoid errors in our computing experiments?		Systematic Error Random Error Errors in Time Measurements
Apr 30	What is the worse case runtime for Horner's method?	HW10 due Fri, Apr 30	Using Maple
15			**Net-centric computing**
May 3	How *does* my email get from here to there?	Dale et al. Ch. 15	History of the Internet Email, telnet, FTP, HTTP Packet switching and sniffing
May 5	Why is Wheaton College the addresses `155.47.--.--`		**Communications and networking** Topologies, protocols
May 7	Is it really possible for someone to listen to my online chats?		Domain Name system
Final		**Final Exam**	Wednesday, May 12, 9am

Exact pages to read and homework exercises to be submitted will be assigned in lecture.

Homework solutions *must* show all your work. Let me say that more directly: do not just submit a homework exercise that shows only your answer. You will *not* get credit for homework problems that do not show *all* your work.

Homework solutions *must* be neat! I know you do not give your English professors "hen-scratch" when you write a paper. No, you write drafts, edit, print, correct, print, and submit a neat final draft. I expect the same in your homework submissions. As you work on the homework, do not concern yourself with how things look, in fact, you should have multiple sheets of scrap paper about as you work on a solution. **BUT**, once you are finished, **you must transcribe your solutions onto a new piece of paper.** Use lots of drawings where appropriate and don't be afraid to write neat notes in the margins that explain your solution procedure. Use many pieces of paper and staple them together. So, I reserve the right to deduct points for sloppy submissions or submissions that are not stapled together, even if the answers are correct.

Honor Code Revisited: It goes without saying that all submitted work will be the student's own, in keeping with the Wheaton Honor Code. For homework, all work must be your own from beginning to end.