MATH/CSCI 243 SYLLABUS

COURSE NAME, NUMBER	Discrete Mathematics with Functional Programming
SEMESTER, YEAR	Fall 2006
PROFESSOR	T. VanDrunen
OFFICE / TELEPHONE / EMAIL	Armerding 112 752-5692 Thomas.VanDrunen@wheaton.edu
OFFICE HOURS	TR 10:30-11:30 am, 1:00-3:30 pm
COURSE WEBSITE	`http://cslab.wheaton.edu/~tvandrun/m243`

RESOURCES VanDrunen, Thomas, *Discrete Mathematics and Functional Programming*, 2006. Available only at the Wheaton College bookstore.

COURSE DESCRIPTION

Sets, logic, the nature of proof, induction, algorithms, algorithm correctness, relations, lattices, functions, and graphs. Functional programming and recursion using the ML programming language.

GOALS AND OBJECTIVES

1. Students will be able to articulate the operations of formal reasoning.
   - Using sets for categorical thinking.
   - Using the rules of logic for deduction and inference.
   - Using relations, functions, and graphs for modeling relationships among categories.
   - Using types in ML for representing sets and functions.
2. Students will be able to compose mathematical proofs.
   - Using direct proof, proof by contradiction, and mathematical induction.
   - Proving properties of sets, correctness of algorithms, properties of relations, properties of functions, and properties of graphs.
3. Students will be able to compose algorithms.
   - Using imperative/iterative techniques.
   - Using a recursive system of functions.
4. Students will be able to compose programs in the ML programming language.

PRIMARY ASSESSMENT PROCEDURES

1. Proof-writing assignments will give students practice in writing proofs on the various topics and help them identify concepts on which they need more work.
2. Program-writing assignments will give students practice in writing programs and implementing algorithms; they will also demonstrate the connection between mathematical propositions and algorithms.
3. Tests and final exam will evaluate students' mastery of these skills.

Grading:

	weight
Homework	25
Test 1	15
Test 2	15
Test 3	15
Final exam	30

SPECIAL EXPECTATIONS

Academic Integrity

Students are encouraged to discuss homework problems and ideas for solutions. However, your solutions, proofs, and programs must be your own. If you are having trouble debugging a program you have written, you may show it to a classmate to receive help; likewise you may inspect a classmate's incorrect program to give help. However, you should not show *correct* code to a classmate, nor should you look at another student's correct code, to give or receive help. Programs on which students have unfairly corroborated will not be accepted.

Late assignments

Late assignments will not be accepted. If you have not completed an assignment on time, hand in what you have completed by the due date for partial credit.

Attendance

While I was an undergraduate, I missed a grand total of two classes, and one of them was to take the GRE. I expect the same from my students. Since being a student is your current vocation, since your learning now will affect your ability to support a family and church later in life, and since you, your family, and/or a scholarship fund are paying a large sum of money to educate you, being negligent in your schoolwork is a sin. I do not take attendance, but I do notice. Unexcused absences will make me less willing to help you during office hours.

Special needs

Whenever possible, classroom activities and testing procedures will be adjusted to respond to requests for accommodation by students with disabilities who have documented their situation with the registrar and who have arranged to have the documentation forwarded to the course instructor. Computer Science students who need special adjustments made to computer hardware or software in order to facilitate their participation must also document their needs with the Registrar in advance before any accommodation will be attempted.

I. Set, symbol, representation
- A. Sets and elements
- B. Expressions and types
- C. Set operations
- D. Tuple and lists

II. Logic
- A. Logical propositions and forms
- B. Conditionals
- C. Argument forms
- D. Predicates and quantifiers
- E. Multiple quantification

III. Proof
- A. Subset proofs
- B. Set equality and empty proofs
- C. Conditional proofs

IV. Algorithm
- A. Algorithms
- B. Induction
- C. Algorithm correctness
- D. Axiomatic semantics
- E. From theorems to algorithms
- F. Analysis of algorithms

V. Relation
- A. Relations
- B. Properties of relations
- C. Closures
- D. Partial orders
- E. Lattices

VI. Function
- A. Functions
- B. Images and inverse images
- C. Properties of functions
- D. Inverse and composition of functions

VII. Functional programming
- A. Recursion
- B. First class functions
- C. Newton's method
- D. Recurrence relations

VIII. Graph
- A. Graphs
- B. Paths and cycles
- C. Isomorphisms