Syllabus: Course Policies and Procedures

Course Information

Course: Discrete Math
Course Number: MATH 218 - xx
Semester:
Classroom:
Day and Time:

Contact Information

Instructor:
Office:
Email:
Phone:
Office Hours:

Course Description

This course is designed to give necessary mathematical background to students in computer science programs. Topics include logic, sets, basic number theory, induction and recursion, counting, relations, and graphs. Prerequisite: C- or higher in MATH217 Precalculus; waiver by placement testing results; or departmental approval.

Required Text and Materials

Discrete Mathematics and Its Applications (Seventh Edition) by Rosen. Published by McGraw-Hill

Teaching Procedures

This course will be taught in a lecture/discussion format with ample opportunity for student questions. Generally, class will begin with a question and answer session on the most recent homework assignment. New material will then be presented in a lecture format and homework be assigned to reinforce the topics covered in class.

Instructional Objectives

COURSE OUTCOMES	SAMPLE OUTCOMES ACTIVITIES	SAMPLE ASSESSMENT TOOLS
Upon successful completion of this course students should:	To achieve these outcomes students may engage in the following activities:	Student learning may be assessed by:
1. Demonstrate an understanding of logic and proofs; (QL)	• Express written statements as statements using propositional logic notation • Use propositional logic to determine truth values of statements • Examine strategies for writing mathematical proofs and for avoiding mistakes in writing proofs	• Homework • In-class problem sets • Quizzes • Exams
2. Demonstrate an understanding of set	• Use set notation to represent a collection of objects • Perform set operations	• Homework • In-class problem sets

notation, matrices, and functions; (QL)	• Examine properties and operations of functions • Develop notation for sequences and summations • Examine notation and operations involving matrices • Describe growth rates of functions using big-O notation	• Quizzes • Exams
3. Demonstrate an understanding of necessary concepts from introductory number theory; (QL)	• Perform operations using modular arithmetic • Convert numbers between different bases • Determine whether values are prime and use the Euclidean algorithm to find the greatest common divisor • Solve congruences	• Homework • In-class problem sets • Quizzes • Exams
4. Prove mathematical statements using induction; (QL)	• Determine when mathematical induction serves as an appropriate proof technique • Perform proofs by induction by constructing appropriate basis and inductive steps • Determine when strong induction or the well-ordering property should be used in proofs • Define functions recursively and perform proofs involving recursively defined functions	• Homework • In-class problem sets • Quizzes • Exams
5. Use counting techniques for problem solving; (QL)	• Use the product, sum, inclusion-exclusion, and division rules to solve counting problems • Use the Pigeonhole Principle to solve counting problems • Determine the difference between combinations and permutations and using each in solving counting problems • Solve applications of recurrence relations	• Homework • In-class problem sets • Quizzes • Exams
6. Demonstrate an understanding of the relationship between elements of sets; (QL)	• Determine when a relation is reflexive, symmetric, antisymmetric, and/or transitive • Represent relations using matrices and/or graphs. • Find equivalence classes	• Homework • In-class problem sets • Quizzes • Exams
7. Demonstrate an understanding of graphs; (QL)	• Examine definitions and terminologies of basic graph theory • Construct graphs • Classify paths and connectivity of graphs • Determine whether a graph has an Euler circuit or an Euler path • Determine whether a graph has a Hamiltonian path	• Homework • In-class problem sets • Quizzes • Exams

This course includes the following core competencies: Quantitative Literacy (QL).

Basis for Student Grading

Grades for this course will be assigned as follows:

Grade	Average
A	93%-100%
A-	90%-92%
B+	87%-89%
B	83%-86%
B-	80%-82%
C+	77%-79%

Grade	Average
C	73%-76%
C-	70%-72%
D+	67%-69%
D	63%-66%
D-	60%-62%
F	0-59%

The grade you earn is the grade you will receive in this course. Grades are not negotiable. You will not be allowed to make up work, substitute alternative assignments, or submit extra assignments in order to improve your grade during the semester or after the semester ends.

Grades of incomplete are given only in situations when extenuating circumstances prevent a student from taking the final exam or fulfilling a specific requirement in the course. The grade of "I" cannot be used to give students additional time to complete course assignments in order to raise their grade.

Basis for Evaluating Student Performance

The grade for this course will be weighted based on the following categories:

- *Exams (70%)*: There will be four in-class exams given throughout the semester, approximately every 3 weeks. Exams must be taken during the regular class time and no make-up exams will be given. The lowest exam grade will be dropped. Your exam average will account for 60% of your final grade.
- *Final Exam (30%)*: The course will culminate in a cumulative final exam. It will be worth 30% of your final grade.

There is no extra credit available for this course.

Tentative Test Schedule/Assignment(s) Schedule

Assignment:	Tentative Date:
Test 1	
Test 2	
Test 3	
Test 4	
Final Exam	

Attendance

Attendance for this course is mandatory. After the third absence, students will lose two points per absence thereafter from their final average. I will take attendance at the beginning of every class, and students not present at that time will be marked absent for the class, even if they show up late. If you must miss a regular class, you are still responsible for the material that was presented in class. The average student needs to attend all class meetings in order to be successful in this course.

Accommodations Statement

The Office of Access & Disability Resources (ADR) is committed to ensuring that students with disabilities have equal, effective, and meaningful access to all academic programs, community events, goods, and services provided by the College in compliance with the Americans with Disabilities Act and Section 504 of the Rehabilitation Act. Students with

disabilities interested in establishing eligibility for services and/or academic accommodations will need to voluntarily disclose their disability to ADR, submit qualifying documentation and complete an ADR Intake meeting. During the meeting reasonable and appropriate academic accommodations will be determined. Visit the Office of Access and Disability Services website for more information and to begin the process to establish eligibility.

ADR uses a platform called Accommodate to manage the accommodation process. To make an initial accommodation request, please submit a request through Accomodate to ADR.

Returning students with accommodations previously approved can request updated Accommodation Letters by accessing the ADR card in the MyMassasoit student portal. Log into Accommodate and click on Semester Request.

Students are expected to check in with faculty regarding approved accommodations. Communication between students, faculty, and ADR is key to successful accommodation implementation.

Students can contact ADR at 508-588-9100 x1807 or by email at adr@massasoit.mass.edu.

Course Outline

Week/Dates	Scheduled Topical Coverage	Other Notes
Week 1	• Course Introduction • Section 1.1 (Propositional Logic) • Section 1.2 (Applications of Propositional Logic)	
Week 2	• Section 1.3 (Propositional Equivalences) • Section 1.7 (Introduction to Proofs) • Section 1.8 (Proof Methods and Strategy)	
Week 3	• Section 2.1 (Sets) • Section 2.2 (Set Operations) • Section 2.3 (Functions)	
Week 4	• Section 2.4 (Sequences and Summations) • **Exam 1**	
Week 5	• Section 2.6 (Matrices) • Section 4.1 (Divisibility and Modular Arithmetic) • Section 4.2 (Integer Representations and Algorithms)	
Week 6	• Section 4.3 (Primes and Greatest Common Divisors) • Section 4.4 (Solving Congruences) • Section 5.1 (Mathematical Induction)	
Week 7	• Section 5.2 (Strong Induction and Well-Ordering) • Section 5.3 (Recursive Definitions and Structural Induction) • Section 5.4 (Recursive Algorithms)	
Week 8	• Section 5.5 (Program Correctness) Optional • **Exam 2**	
Week 9	• Section 6.1 (The Basics of Counting) • Section 6.3 (Permutations and Combinations) • Section 8.1 (Applications of Recurrence Relations)	
Week 10	• Section 8.2 (Solving Linear Recurrence Relations) • Section 9.1 (Relations and Their Properties) • Section 9.2 (*n*-ary Relations and Their Applications)	
Week 11	• Section 9.3 (Representing Relations) • **Exam 3**	
Week 12	• Section 9.4 (Closures of Relations) • Section 10.1 (Graphs and Graph Models)	
Week 13	• Section 10.2 (Graph Terminology and Special Types of Graphs) • Section 10.3 (Representing Graphs and Graph Isomorphism)	
Week 14	• Section 10.4 (Connectivity) • **Exam 4**	

Week 15	• Section 10.5 (Euler and Hamilton Paths) • **Review for final**	
	• **Final Exam Period**	