Math 272 Course Content and Objectives

COURSE CONTENT AND SCOPE - **Lecture:** Outline the topics included in the lecture portion of the course *(Outline reflects course description, all topics covered in class).*	Hours Per Topic	COURSE OBJECTIVES - **Lecture:** Upon successful completion of this course, the student will be able to...*(Use action verbs - see Bloom's Taxonomy for 'action verbs requiring cognitive outcomes.')*
Sentential logic: Logical connectives, truth tables, variables and sets, operations on sets, and conditionals and biconditionals. Formal logic including tautologies, propositional logic, logic programming, predicates, and predicate logic, logic networks, and minimization. Recursive definitions, recurrence relations, and analysis of algorithms. Boolean algebras.	5	Construct a truth table. Prove statements about sets. Apply the conditional and contrapositive laws.
Quantificational logic: Quantifiers, equivalences involving quantifiers, and additional operations on sets.	5.5	Analyze statements with quantifiers. Apply the quantifier negation laws. Prove statements about the power set.
Proofs: Proofs involving negations, conditionals, quantifiers, conjunctions, biconditionals, and disjunctions. Existence and uniqueness proofs. Epsilon-delta proofs.	8	Apply the proof strategies for statements with negation, conditionals, quantifiers, conjunctions, and disjunctions. Prove existence and uniqueness statements. Write proofs using symbolic logic and Boolean algebra.
Relations: Ordered pairs and Cartesian products, binary relations, relations, ordering relations, closures, and equivalence relations. Graph algorithms including directed graphs.	7	Construct the Cartesian product of two sets. Identify the domain and image of a relation. Identify a partial order relation.
Functions: Functions, one-to-one and onto functions, inverses of functions, images, and inverse images.	5	Construct and identify functions. Determine when a function is one-to-one or onto. Identify the image and inverse image.
Mathematical induction: Proof by mathematical induction, recursion, recursion relations, recurrance relations, recursive definitions, strong induction, and additional information on closures. Application to the correctness of algorithms.	7	Prove a statement using mathematical induction or strong induction. Use recursion to prove statements. Solve a recursion relation.
Infinite sets: Equinumerous sets, the Pigeonhole Principle, countable and uncountable sets, the Cantor-Schroder-Bernstein theorem.	4	Identify when sets are equinumerous by constructing a one-to-one and onto function between them. Prove statements using the Pigeonhole Principle.
Counting and combinatorics: Sequences, permutations, the number of ordered subsets, the number of ordered subsets of a given size, inclusion-exclusion, the binomial theorem, Pascal's triangle, and Fibonacci numbers.	12	Compute combinations and permutations using binomial coefficients. Use sets to solve problems in combinatorics and probability theory.
Elementary number theory: Divisibility, modular arithmetic, primes, testing if a number is prime, Fermat's little theorem, the Euclidean algorithm, congruences, and modeling arithmetic.	13	Compute congruences. Prove Fermat's Little Theorem and the Euclidean algorithm.
Graphs and trees: Computer networks, graphs, representations of graphs, path, cycles, articulation points, connectivity, Eulerian walks, Hamiltonian circuits, trees, decision trees, minimal spanning tree,	14	Prove statements about the connectivity of a graph. Prove statements about the subgraph of a graph. Apply matrices to analyze graphs and trees.

shortest path, Warshall's algorithm, finding the best tree, and the Traveling Salesman problem.		
Coloring maps and Graphs: Coloring regions and graphs with two colors, coloring graphs with many colors, and the Four Color Theorem.	5	Prove statements about the coloring of maps.
Computer topics, complexity and cryptography: Computation, tree traversal algorithms, analysis of algorithms, matrices including relations and databases, languages including algebraic structures, finite state machines, formal languages, Huffman Codes, computer logic including Boolean algebra structure, verifying a password without learning it, and public key cryptography.	2.5	Use finite state machines to model computer operations. Describe the public key cryptography process.
Final examination.	2	Final examination.
Total:	90	
Total Lecture Hours In Section I Class Hours:	90	