

**Concepts and Skills for CMIS 160, Discrete Mathematics for Computing**  
(Revised 26 January 2005)

Topic	Concepts	Skills
Sets and Functions	Set Subset Empty Set Union, Intersection Difference, Complement Cartesian Product “Set Identities” Power Set Function Domain Co-domain Special functions: <i>div</i> and <i>mod</i> , <i>floor</i> and <i>ceiling</i> [Direct] Image Inverse Image One-to-One (Injective) Onto (Surjective) Bijection Inverse Pigeonhole Principle	Find the subset of a set consisting of the elements satisfying a condition. Find unions, intersections, differences, and complements of sets. Apply set identities. Find the power set of a set . Find the image of a function. Determine whether a function is one-to-one. Determine whether a function is onto. Determine whether a function is a bijection. If a function is a bijection, find its inverse.

<b>Topic</b>	<b>Concepts</b>	<b>Skills</b>
Relations	Binary Relation Function as a type of Relation Special relations: divisibility, congruence Modulo n Inverse of a Relation Directed Graph n-ary Relation Reflexive Symmetric Transitive Equivalence Relation Partition Equivalence Class Partial Order Antisymmetric Total Order	Given a relation presented in one form (set of ordered pairs, directed graph, rule), present it in other forms. Given a relation, determine which of the following properties apply to it: reflexive, symmetric, transitive, antisymmetric, partial order, total order. Given a relation, determine whether it is an equivalence relation. Find the equivalence classes of an equivalence relation. Determine whether two numbers are congruent modulo n.
Propositional Logic	Proposition Logical operators (connectives) Truth Table Converse Inverse Contrapositive	Perform a truth-table analysis on a formula Recognize the various verbal forms of a conditional statement, and restate any of them in if...then form. Solve a logical word-problem using the three-step process (construct model, analyze model, interpret results) Construct the negation of statements that include and, or, not, and implies. Construct the inverse, converse, and contrapositive of a conditional statement.
Logic Circuits	Gate (AND, OR, NAND, NOR, NOT, XOR) Circuit Feedback	Write the logical formula corresponding to a simple circuit Draw a circuit corresponding to a simple formula
Predicate Logic	Set Function Propositional function (predicate) Universal Quantifier Existential Quantifier	Translate a fully verbal statement into a hybrid formal/verbal form in which all the quantifiers (and their associated variables) are explicitly displayed. Translate a hybrid formal/verbal statement into a fully verbal statement. Restate the negation of a statement with quantifiers by “moving the negation operator inside the quantifiers.”
Proof	Proof Counterexample Direct Proof Indirect Proof	Prove simple statements about numbers by working directly from definitions of the concepts involved in the statements. Read a simple proof and determine whether it is valid or invalid. If it is invalid, spot the error.

Topic	Concepts	Skills
Sequences and Mathematical Induction	Sequence $\Sigma$ notation $\Pi$ notation Dummy variable Factorial Mathematical Induction	Compute the n-th term of a sequence, given a formula. Expand a $\Sigma$ expression into a sum of terms Write a sum of terms as a $\Sigma$ expression Expand a $\Pi$ expression into a product of terms Write a product of terms as a $\Pi$ expression Accomplish a change of variable in a $\Sigma$ or $\Pi$ expression Compute the factorial of a positive integer Construct simple proofs using mathematical induction
Recursion	Recursively Defined Sequence	Determine whether a proposed formula for the n-th term of a recursively defined sequence is correct. Use a spreadsheet to compute a good number of terms of a recursively defined sequence, and graph them.
Introduction to Graphs	Graph Edge Vertex Endpoint Loop Adjacency Incidence Isolated Vertex Directed Graph Simple Graph Complete Graph Complete Bipartite Graph	Given a graph presented graphically, identify the edges, the vertices, and the endpoints of the edges. Determine which vertices in a graph are adjacent to which. Determine which edges of a graph are incident to which vertices. Identify loops and parallel edges in a graph. Locate all isolated vertices of a graph. Determine whether a graph is simple. Determine whether a graph is a complete graph. Determine whether a graph is a complete bipartite graph.

Topic	Concepts	Skills
Additional Topics on Graphs	Degree Total Degree Walk Path Simple Path Closed Walk Circuit Simple Circuit Connected Graph Connected Component Euler Circuit Euler Path Hamiltonian Circuit Traveling Salesman Problem Isomorphism of Graphs	Compute the degree of a vertex. Compute the total degree of a graph. Examine an alternating sequence of vertices and edges and determine whether it is a walk, path, simple path, closed walk, circuit, simple circuit, Euler circuit, Euler path, or Hamiltonian circuit. Determine whether a graph is connected, and if not, identify its components. Determine whether a graph has an Euler circuit. If a graph has an Euler circuit, find one. Recognize differences in two graphs that would establish that the graphs are not isomorphic. Explain what the traveling salesman problem is
Trees	Tree Terminal Vertex Leaf Internal Vertex Rooted Tree Level Height (of a Rooted Tree) Child Parent Sibling Ancestor Descendant Binary Tree Full Binary Tree Weighted Graph Spanning Tree Minimal Spanning Tree	Determine whether a graph is a tree by inspection. Determine whether a graph is a tree by comparing the number of vertices with the number of edges. Given the number of vertices in a tree, find the number of edges, and vice versa. Determine whether a tree is a binary tree. Determine whether a tree is a full binary tree. Find a spanning tree of a graph. Find a minimal spanning tree of a weighted graph.

<b>Topic</b>	<b>Concepts</b>	<b>Skills</b>
Languages/Automata	Formal Language Grammar Regular Grammar Regular Expression Finite-State Automaton Initial State Accepting State Transition Diagram State Table	Given a transition diagram for a finite-state automaton, construct its state table. Given a state table for a finite-state automaton, draw its transition diagram. Given a finite-state automaton and a string, determine whether the string is accepted by the automaton. Find the language defined by a regular expression Write a regular expression defining a language