Douglas College

EFFECTIVE: MAY 2006
CURRICULUM GUIDELINES

A.	Division:	Instructional	Effective Date:		May 2006	
B.	Department / Program Area:	Mathematics Faculty of Science & Technology	Revision	X	New Course	
			If Revision, Section(s) Revised:		F, G, H, J, M, N, O, P, Q	
			Date of Previous Revision:		September 2004	
			Date of Current Revision:		May 24, 2005	

C: MATH 1130 **D**: Discrete Mathematics I **E:** 3

Subject & Course No.	Descriptive Title	Semester Credits

F: Calendar Description:

This is the first of two Discrete Mathematics courses for Computing Science students. Topics include logic, set theory, counting, functions, relations, graphs, trees, finite state machines and formal languages.

G: Allocation of Contact Hours to Type of Instruction / Learning Settings

Primary Methods of Instructional Delivery and/or Learning Settings:

Lecture and tutorials

Number of Contact Hours: (per week / semester for each descriptor)

4 – Lecture 1 - Tutorial

Number of Weeks per Semester:

15

H: Course Prerequisites:

BC Principles of Mathematics 12 (C grade or better) or equivalent

I: Course Corequisites:

None

J: Course for which this Course is a Prerequisite

MATH 2230 and MATH 2232

K: Maximum Class Size:

35

L: PLEASE INDICATE:

- [ ] Non-Credit
- [ ] College Credit Non-Transfer
- [X] College Credit Transfer:

SEE BC TRANSFER GUIDE FOR TRANSFER DETAILS (www.bctransferguide.ca)

M: Course Objectives / Learning Outcomes

At the end of the course, the successful student should be able to:

- write English statements in symbolic form using prepositional variables or functions, logical connectives and any necessary quantifiers;
- determine the truth value of a statement under an interpretation;
- determine the negation, converse or contrapositive of a statement;
- verify logical equivalencies;
- demonstrate an understanding of tautologies, contradictions and duals;
- prove the properties of logic;
- determine the cardinality of sets, subsets, power sets and Cartesian products;
- combine sets using the set operators;
- prove set identities by showing that each expression is a subset of the other;
- use membership tables or Venn diagrams to prove set identities;
- classify functions as injective, surjective or bijective;
- demonstrate an understanding of domains, codomains, ranges, mappings and images;
- create new functions by composition;
- find the inverse of an injective function;
- demonstrate an understanding of the floor and ceiling functions;
- compute finite sums;
- determine if a set is countable or uncountable;
- give a big-O estimate for a function;
- write a simple algorithm in pseudocode;
- determine the time complexity of simple algorithms;
- demonstrate an understanding of divisibility, the greatest common divisor and modular arithmetic;
- use the Euclidean algorithm to find the gcd of two numbers;
- convert between binary, octal and hexadecimal;
- find the sum, difference, product, join, and meet of two matrices;
- demonstrate an understanding of the rules of inference;
- analyze an argument as to its validity using the concepts of mathematical logic;
- use a direct proof, indirect proof, or contradiction to prove a mathematical theorem;
- prove mathematical theorems using formal inductive techniques;
- give a recursive definition of a function or a set;
- use the sum and product rules and tree diagrams to solve basic counting problems;
- apply the inclusion-exclusion principle to solve counting problems for two tasks;
- solve counting problems using the Pigeon-Hole Principle;
- count unordered selections of distinct objects;
- count ordered arrangements of objects of a finite set;
- find the expansion of a binomial;
- determine the probability of a combination of events for an equi-probable sample space;
- determine whether or not a relation is reflexive, symmetric, and or transitive;
- combine relations and form the composite of two or more relations;
- find the inverse and complement of a relation;
- determine the projection and join of two n-ary relations;
- represent a relation as a matrix and a digraph;
- find the reflexive, symmetric and transitive closures of a relation;
- identify the various types of graphs;
- draw graph models;
- demonstrate an understanding of the vocabulary of graph theory;
- determine whether a graph is bi-partite or not;
- represent a graph as an adjacency matrix and an incidence matrix;
- determine whether a pair of graphs are isomorphic;
- find circuits and paths in a graph;
- distinguish between a graph and a tree;
- describe the components and properties of various types of trees;
- determine whether a string belongs to the language generated by a given grammar;

- classify a grammar;
- find the language created by a grammar;
- draw the state diagram for a finite-state machine;
- construct a finite-state machine to perform a function;
- determine the output of a finite state machine;

N: Course Content:

1. Logic
2. Set Theory
3. Functions
4. Algorithms, Integers and Matrices
5. Mathematical Reasoning and Recursive Definitions
6. Counting
7. Relations
8. Graphs and Trees
9. Languages and Finite State Machines

O: Methods of Instruction

Lectures, problem sessions, tutorial sessions and assignments

P: Textbooks and Materials to be Purchased by Students

Rosen, H.R., Discrete Mathematics and Its Applications, 5th Edition, McGraw Hill, 2003.

Q: Means of Assessment

Evaluation will be carried out in accordance with Douglas College policy. The instructor will present a written course outline with specific evaluation criteria at the beginning of the semester. Evaluation will be based on some of the following:

1.	Weekly tests	0 – 40 %
2.	Midterm tests	20 – 70%
3.	Assignments	0 – 15%
4.	Attendance	0 – 5%
5.	Class participation	0 – 5%
6.	Tutorials	0 – 10%
7.	Final examination	30 – 40 %

R: Prior Learning Assessment and Recognition: specify whether course is open for PLAR

None

Course Designer(s) Natasha Davidson

Education Council / Curriculum Committee Representative

Dean / Director Des Wilson

Registrar Trish Angus