



## EFFECTIVE: SEPTEMBER 2004 CURRICULUM GUIDELINES

**A.** Division: **Instructional** Effective Date: **September 2004**

**B.** Department / **Mathematics** Revision  New Course   
 Program Area: **Faculty of Science & Technology** If Revision, Section(s) **C, H**  
 Revised:  
 Date of Previous Revision: **May 1997**  
 Date of Current Revision: **September 2004**

**C: MATH 2230** **D: Discrete Mathematics II** **E: 3**

Subject & Course No.	Descriptive Title	Semester Credits						
<p><b>F:</b> Calendar Description:</p> <p>This is the second of two Discrete Mathematics courses for Computing Science students. Topics include complexity of algorithms, recursion, recurrence relations, generating functions, equivalence relations, partial orders, partitions, graphs and trees, cycles and paths, shortest-path algorithms, minimal spanning trees, tree traversal and applications of trees and graphs.</p>								
<p><b>G:</b> Allocation of Contact Hours to Type of Instruction / Learning Settings</p> <p>Primary Methods of Instructional Delivery and/or Learning Settings:</p> <p style="text-align: center;">Lecture</p> <p>Number of Contact Hours: (per week / semester for each descriptor)</p> <p style="text-align: center;">4 hours per week</p> <p>Number of Weeks per Semester:</p> <p style="text-align: center;">15</p>	<p><b>H:</b> Course Prerequisites:</p> <p style="text-align: center;">MATH 1130</p>							
	<p><b>I:</b> Course Corequisites:</p> <p style="text-align: center;">None</p>							
	<p><b>J:</b> Course for which this Course is a Prerequisite</p> <p style="text-align: center;">None</p>							
	<p><b>K:</b> Maximum Class Size:</p> <p style="text-align: center;">35</p>							
<p><b>L:</b> PLEASE INDICATE:</p> <table style="border-collapse: collapse;"> <tr> <td style="border: 1px solid black; width: 30px; height: 20px;"></td> <td>Non-Credit</td> </tr> <tr> <td style="border: 1px solid black; width: 30px; height: 20px;"></td> <td>College Credit Non-Transfer</td> </tr> <tr> <td style="border: 1px solid black; width: 30px; height: 20px; text-align: center;">X</td> <td>College Credit Transfer:</td> </tr> </table> <p style="text-align: center;">SEE BC TRANSFER GUIDE FOR TRANSFER DETAILS (<a href="http://www.bccat.bc.ca">www.bccat.bc.ca</a>)</p>				Non-Credit		College Credit Non-Transfer	X	College Credit Transfer:
	Non-Credit							
	College Credit Non-Transfer							
X	College Credit Transfer:							

**M:** Course Objectives / Learning Outcomes

The student should be able to:

- determine whether a set is countable or uncountable;
- demonstrate Cantor's diagonalization process;
- devise recursive algorithms and compare with iterative algorithms;
- use a loop invariant to prove that a program segment is correct;
- determine the worst-case and average-case complexity of a simple algorithm;
- determine the number of ordered and unordered selections of  $r$  elements chosen with and without repetition from a set with  $n$  elements;
- determine the permutations of sets with indistinguishable objects;
- enumerate the ways distinguishable objects can be placed into distinguishable boxes;
- develop an algorithm to generate permutations of a set;
- develop a recurrence relation to model a problem;
- solve recurrence relations iteratively;
- solve linear homogeneous recurrence relations with constant coefficients of degree two;
- verify solutions to linear inhomogeneous recurrence relations;
- determine the big-O of divide-and-conquer recurrence algorithms such as the binary search;
- apply the inclusion-exclusion principle to problems with more than two sets;
- use the principle of inclusion-exclusion to solve counting problems modeled after the problem of finding the number of integer solutions of a linear equation with constraints;
- solve counting problems modeled after the number of onto functions from one finite set to another;
- count the number of derangements of a set and solve counting problems based on this principle;
- derive generating functions for a sequence;
- use ordinary and exponential generating functions to solve counting problems;
- use a generating function to solve a recurrence relation;
- determine if a relation is an equivalence relation;
- determine the equivalence classes of an equivalence relation;
- determine if a collection of subsets is a partition of a given set;
- demonstrate an understanding of partial order, total order, lexicographic order, well-ordered, maximal element, minimal element, greatest element, least element, bounds, lattice;
- draw a Hasse diagram of a poset;
- represent a digraph as an adjacency matrix or incidence matrix;
- determine whether a pair of graphs are isomorphic;
- demonstrate an understanding of connected, simple path, weighted graph, circuit, subgraph, cut vertices, cut edges and degree of a vertex;
- determine whether a graph has an Euler path or circuit;
- determine whether a graph has a Hamiltonian path or cycle;
- model and solve problems with a graph;
- use Dijkstra's algorithm to find the shortest path between pairs of vertices in a weighted graph;
- demonstrate an understanding of the terminology of rooted trees and  $m$ -ary trees;
- form a binary search tree using a recursive algorithm and analyze the algorithm;
- model a problem using a decision tree;
- construct a binary tree which represents a given prefix coding scheme;
- find the word represented by a bit string given a coding sequence;
- demonstrate an understanding of a Huffman tree;
- demonstrate an understanding of three algorithms for traversing all vertices of an ordered rooted tree (preorder traversal, inorder traversal, postorder traversal);
- represent a complicated expression using a binary tree and write the expression in prefix, postfix or infix notation;
- evaluate a prefix, postfix or infix expression;
- demonstrate an understanding of several sorting algorithms and their complexity (bubble sort, merge sort, selection sort and quick sort);
- find all the non-isomorphic spanning trees of a graph;
- use a depth-first or breadth-first search to produce a spanning tree;
- use Prim's algorithm or Kruskal's algorithm to construct a minimum spanning tree for a weighted graph.

**N:** Course Content:

1. Infinite Sets, Computability and Recursion
  - 1.1. Cardinality of infinite sets
  - 1.2. Recursion and iteration
  - 1.3. Complexity of algorithms
  - 1.4. Program correctness
2. Advanced Counting
  - 2.1. Permutations and combinations with repetition
  - 2.2. Indistinguishable and distinguishable objects
  - 2.3. Recurrence relations
  - 2.4. Solving first and second order linear recurrence relations
  - 2.5. Generating functions
  - 2.6. Solving recurrence relations using generating functions
  - 2.7. Solving counting problems using generating functions
  - 2.8. Divide-and-conquer relations
  - 2.9. Applications of inclusion-exclusion
3. Relations
  - 3.1. Equivalence relations and partitions
  - 3.2. Partial orderings and Hasse diagrams
4. Graphs
  - 4.1. Representations
  - 4.2. Connectivity
  - 4.3. Euler and Hamilton paths
  - 4.4. Shortest path problems
5. Trees
  - 5.1. Applications
  - 5.2. Tree traversals
  - 5.3. Trees and sorting
  - 5.4. Spanning trees
  - 5.5. Minimum spanning trees

**O:** Methods of Instruction

Lectures, problem sessions, and assignments

**P:** Textbooks and Materials to be Purchased by Students

Rosen, H.R., Discrete Mathematics and Its Applications, McGraw Hill, 1995.

**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. Final examination   | 30 %     |

**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